

Руководство

по использованию модуля

«Всплывающие окна»

1 марта 2015г.

Содержание

Введение	3
Назначение модуля, сфера применения	3
Почему лучше использовать данный модуль?	4
Технические требования	4
Загрузка модуля на сайт	4
Сравнение методов создания всплывающих окон	6
Установка статического всплывающего окна	7
Установка динамического всплывающего окна	9
Расположение компонента внутри других	9
Параметры всплывающих окон	10
Описание API модуля	14
Встроенные схемы оформления	16
Примеры установки и настройки	17
Возможные проблемы и их решения	28

Введение

Модуль «Всплывающие окна» предназначен для быстрого создания всплывающих окон на сайте, работающем под управлением 1С-Битрикс.

Данный модуль, начиная с версии 1.0.19, стал платным, но Вы можете устанавливать его на свой сайт из 1С-Битрикс:Маркетплейс для тестирования в течение 14 дней. В режиме тестирования модуль не имеет никаких ограничений, кроме срока действия: после окончания указанного срока действия, модуль перестанет работать.

Настоящее руководство предназначено для лиц, устанавливающих модуль на сайт и настраивающих его для корректной работы: для программистов, верстальщиков, контент-менеджеров, владельцев проектов и др.



Назначение модуля, сфера применения

Изначально модуль был предназначен лишь для создания всплывающих окон по клику на ссылку. С развитием модуля он получил дополнительные возможности, и на данный момент (версия 1.0.23) модуль имеет встроенные функции для создания следующих типов всплывающих окон:

1. Стандартные всплывающие окна
Всплывающее окно содержит текст, изображения, таблицы, ссылки, списки и др.; открывается по клику на ссылку, кнопку, изображение, блок и др.
2. Всплывающие окна с видео
Всплывающее окно содержит видео, размещенное либо с помощью тега `<iframe>`, либо с помощью вставки тегами `<object>` и `<embed>`.
3. Всплывающие окна с картинкой
С помощью всплывающих окон, создаваемых модулем, легко реализовать простейшую галерею фотографий, в которой по клику на уменьшенную фотографию появляется увеличенная.
4. Всплывающие окна с отдельными элементами страниц
Во всплывающем окне возможно разместить такие элементы страниц, как: форма обратной связи, форма заказа обратного звонка, форма авторизации, форма подписки, окно выбора города/местоположения, таблица размеров одежды и др.
5. Всплывающие окна с AJAX-загрузкой
Модуль имеет несколько функций для работы с AJAX, благодаря чему имеется возможность достаточно просто реализовать загрузку данных только по мере необходимости, что позволяет не замедлять открытие страницы предварительной загрузкой содержимого всплывающих окон. В случае загрузки данных для форм это поможет в борьбе со СПАМом. AJAX-загрузка форм дает колоссальные возможности для создания всплывающих форм любого типа и назначения.
6. Всплывающие окна с рекламой
В одном из обновлений (1.0.3) реализовано автоматическое открытие всплывающего окна при загрузке страницы, с возможностью указать задержку по времени между загрузкой страницы и открытием всплывающего окна. Все это позволяет использовать модуль для быстрого и удобного создания баннеров и рекламных объявлений.
7. Всплывающие фиксированные окна, которые видны вне зависимости от прокрутки страницы.

Почему лучше использовать данный модуль?

Мы рекомендуем Вам использовать наш модуль, вместо того, чтобы разрабатывать свою систему всплывающих окон, потому что модуль «всплывающие окна»:

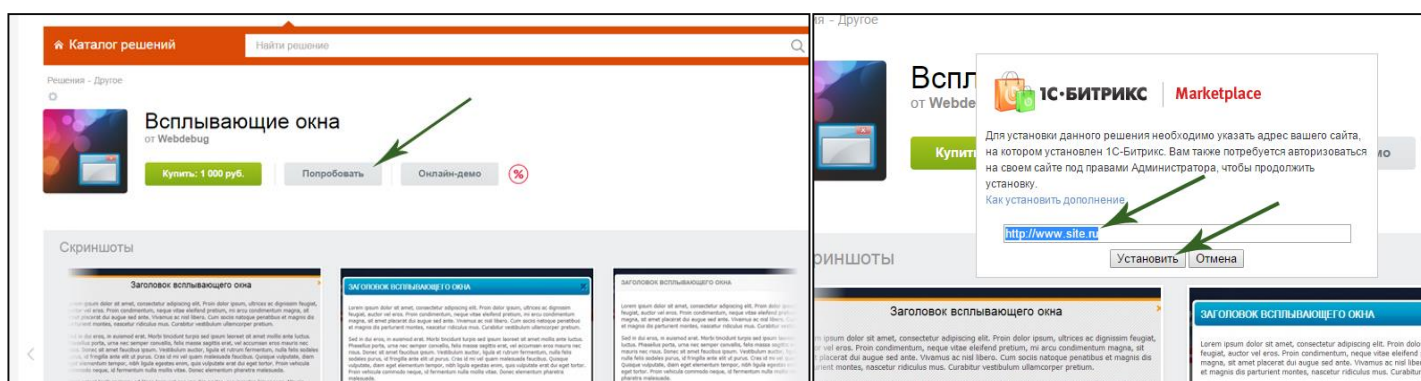
1. максимально прост в установке и использовании,
2. имеет несколько готовых вариантов оформления (в версии 1.0.23 – 11 совершенно разных тем оформления),
3. имеет бесплатную техническую поддержку, которая оперативно помогает в решении проблем,
4. постоянно развивается и улучшается.

Технические требования

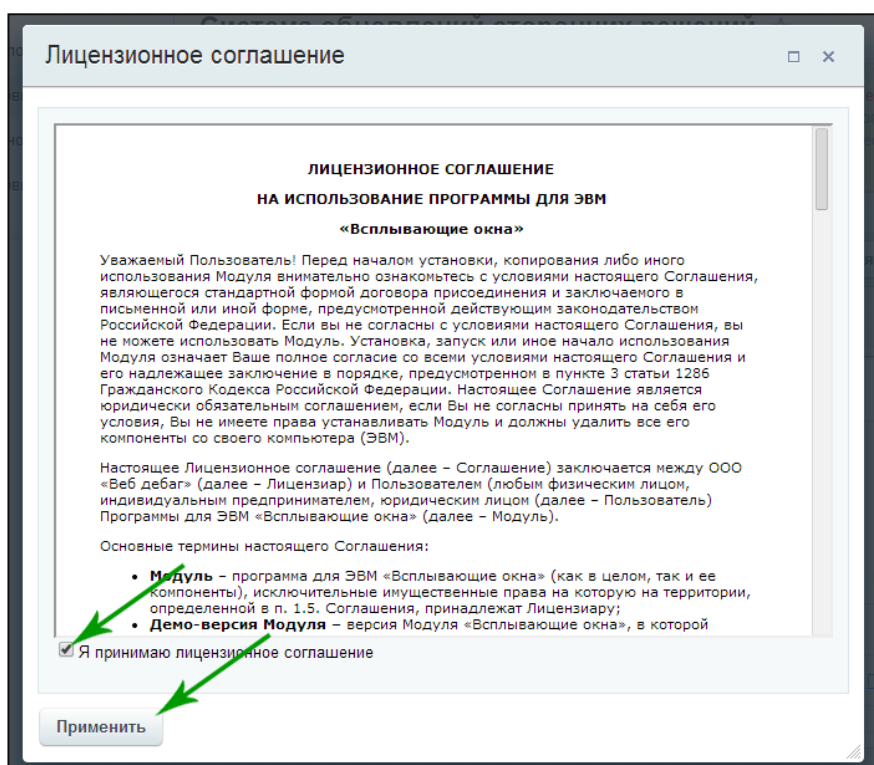
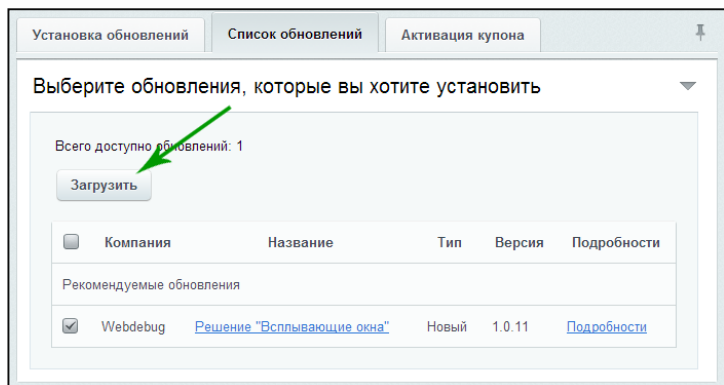
1. Система управления 1С-Битрикс любой редакции, в т.ч. «Первый сайт»,
2. Наличие подключенной на сайте JavaScript-библиотеки jQuery версии не ниже 1.4, либо же отсутствие подключения какой-либо из версий jQuery,
3. Отсутствие JavaScript-ошибок при загрузке страниц сайта.

Загрузка модуля на сайт

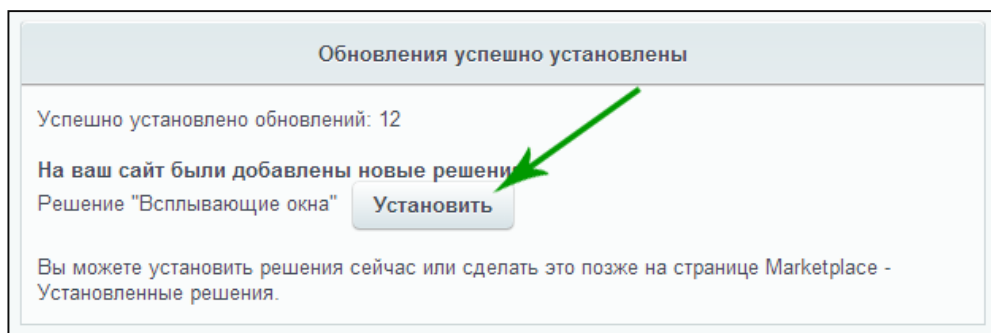
Установка модуля начинается с загрузки его на сайт. Загрузить модуль можно на его [странице](#) в сервисе [1С-Битрикс: Маркетплейс](#). Для этого нажмите кнопку «Установить», в открывшемся окне укажите адрес сайта, например, «<http://www.site.ru>» нажмите «Установить»:



После этого откроется страница административного раздела Вашего сайта, где Вам предлагается загрузка указанного модуля. Нажмите кнопку «Загрузить», в появившемся окне с лицензионным соглашением отметьте галочкой свое согласие и нажмите «Применить»:



После загрузки модуля будет предложено его установить. Нажмите кнопку «Установить»:



После установки откроется список установленных модулей, в числе которых должен быть модуль «Всплывающие окна» со статусом «Установлен»:

	Название	Разработчик	Версия	Дата обновления	Обновления до	Статус
☰	Всплывающие окна (webdebug.popup) Модуль для быстрого создания всплывающих окон	Веб-студия «Webdebug» [ООО «Веб дебаг»]	1.0.11	24.03.2014		Установлен

Примечание: в случае использования модуля в демонстрационном режиме (если лицензия на модуль еще не была активирована), напротив данного модуля (в списке модулей) будет показываться соответствующая информация.

Сравнение методов создания всплывающих окон

Установка добавляет в систему единственный компонент – «Всплывающее окно», который реализует простой механизм создания всплывающего окна (**статическое окно**). «Статическое» означает, что содержимое окна задается администратором (контент-менеджером, программистом и т.д.) вручную, т.е. то, что будет помещено в контент – то и будет отображаться. Также доступна возможность создания всплывающего окна динамическим способом (**динамическое окно**). Это означает, что в окне может содержаться контент в зависимости от некоторых условий. Например, для каждого товара свое содержимое.

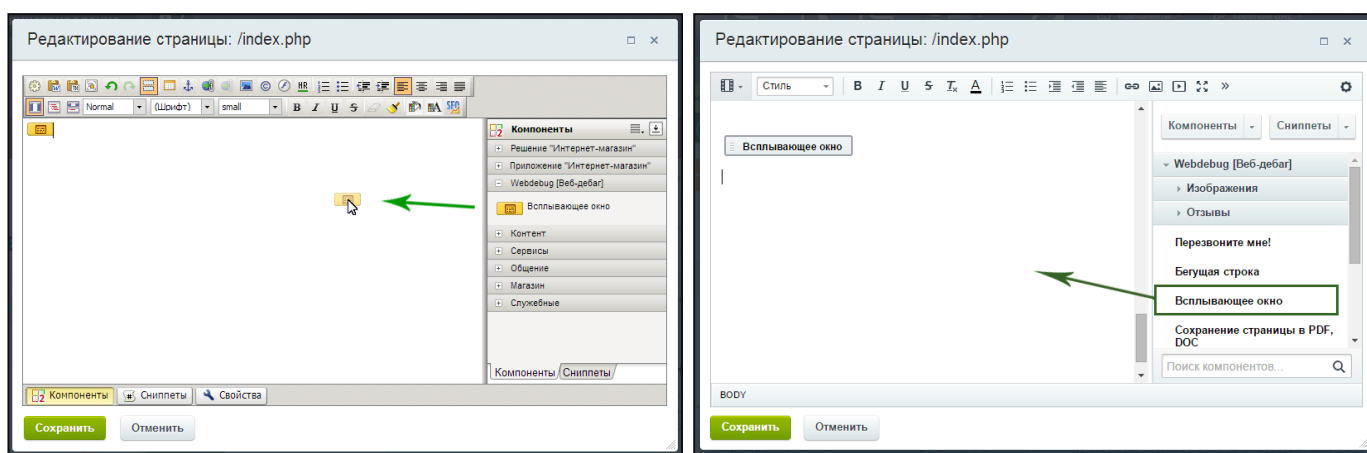
Несмотря на то, что оба метода используют один и тот же внутренний механизм, статический и динамический методы создания всплывающих окон различаются между собой.

	Статическое окно	Динамическое окно
Способ установки всплывающего окна	Устанавливается компонентом «Всплывающее окно»	Устанавливается PHP-кодом
Содержимое всплывающего окна	Загружается из PHP-файла	Задается вручную
Относительная сложность установки	Легко	Сложно

Установка статического всплывающего окна

Статическое всплывающее окно создается с помощью компонента «Всплывающее окно». Просто откройте на странице сайта визуальный редактор и поместите компонент в поле с текстом одним из способов:

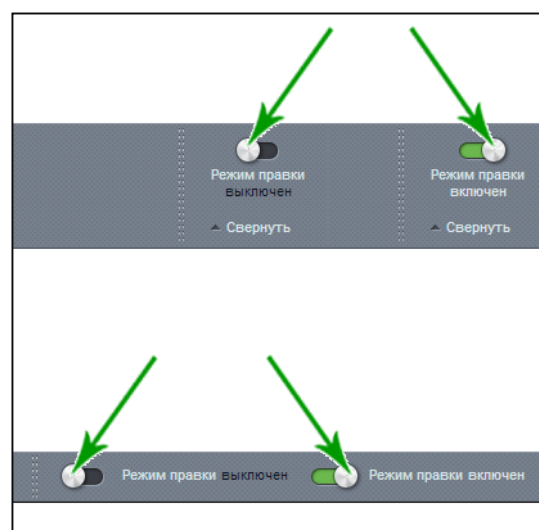
- перенести картинку компонента из правой панели,
- дважды щелкнуть по картинке компонента в правой панели.



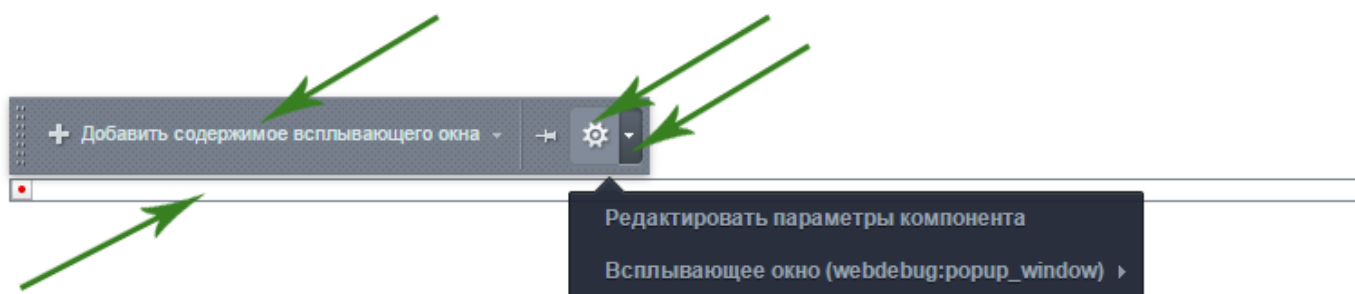
Настройки компонента можно производить, выполнив двойной щелчок на иконке компонента в области текста, но лучше это делать после сохранения, прямо на странице сайта в режиме правки.

После установки компонента нажмите кнопку «Сохранить». Страница обновится, на ней уже располагается компонент «всплывающее окно». Оно пока не работает по одной причине: не задано его содержимое.

Для задания содержимого прежде всего необходимо **перейти в режим правки**. Режим правки задает переключатель, который находится вверху справа на панели управления над сайтом. Щелчок по переключателю или перенос ползунка переключает режим правки между состояниями «включен» и «выключен».



Для управления окном необходимо в режиме правки навести курсором на область с окном (без кнопки открытия эта область является узкой полоской, которую можно определить, передвигая курсор вверх-вниз примерно над этой областью: при наведении она подсвечивается), при этом над компонентом появляется **всплывающая контекстная панель**, на которой расположены кнопка «Добавить содержимое всплывающего окна» (если область уже задана, кнопка будет называться «Изменить содержимое всплывающего окна») и **кнопка, открывающая настройки компонента** (в виде шестеренки), имеющая **выпадающим подменю**, которое открывается по щелчку на значок треугольника справа.



Содержимое статического окна, как уже было указано, загружается из файла. Поэтому компонент для загрузки содержимого всплывающего окна подключает PHP-файл. В работе этого функционала **компонент абсолютно идентичен стандартному битрикс-компоненту** «включаемая область». Настройки компонента позволяют подключать такой файл тремя способами, в зависимости от параметра «Показывать включаемую область»:

1. загрузка из указанного файла, указанного вручную: параметру «Показывать включаемую область» задать значение «из файла», а параметру «Путь к файлу области» - указать, например, «/include/рорип.php» (без кавычек). В этом случае содержимое окна всегда будет одним и тем же, вне зависимости от того, на какой странице открывается окно. Это наиболее распространенный вариант.
2. Загрузка из файла, относящегося к разделу: параметру «Показывать включаемую область» задать значение «для раздела», а параметру «Суффикс имени файла включаемой области» - указать, например, «рорип» (без кавычек). При этом содержимое файла будет сохраняться в файл «sect_inc.php». В таком случае для всех подразделов и страниц этого раздела будет показываться всплывающее окно с одним и тем же содержимым, для других разделов – другое содержимое. Это полезно использовать, например, если необходимо для основных разделов сайта (напр., «Каталог», «Услуги», «О компании» и др.) показывать всплывающие окна с различным содержимым. **Это не относится к разделам каталога, работающем на инфоблоках, где разделом считается раздел инфоблока, а не раздел (папка) с файлами на сервере.**
3. Загрузка из файла, относящегося к странице: аналогично предыдущему пункту, но содержимое всплывающего окна будет разным для каждой страницы, даже в рамках одного раздела. **Внимание!** Начиная с версии модуля 1.0.23, изменен суффикс страницы по умолчанию на «рорип» (ранее был «inc»). Т.е. теперь новые файлы всплывающих окон будут называться (в зависимости от имени файла), например «index_рорип.php».

Во всех случаях, когда указанный настройками файл (по любому из трех вариантов) не создан, компонент предлагает его создать (кнопка «Добавить содержимое всплывающего окна»), если же создан – кнопка «Изменить». **В случае отсутствия указанного файла ни всплывающее окно, ни кнопка для его открытия не будут показаны!** Поэтому, для показа окна необходимо после размещения компонента навести курсором на область компонента и нажать «Добавить содержимое всплывающего окна», в открывшемся окне с визуальным редактором заполнить контент и нажать «Сохранить».

Установка динамического всплывающего окна

Динамическое всплывающее окно устанавливается сложнее, чем статическое, т.к. для установки необходимо редактировать исходный код сайта, и даже предполагает хотя бы начальное знание PHP, JavaScript, jQuery, HTML+CSS. По сути, установка динамического окна – это использование API модуля.

Пример кода для установки динамического всплывающего окна:

```
<?if(CModule::IncludeModule('webdebug.popup')):?>
  <?CWD_Popup::Init(true);?>
  <?$arParams = array(
    'ID' => 'mywindow',
    'NAME' => 'Заголовок окна',
    'WIDTH' => '500',
    'CLASSES' => array('wd_popup_style_05'),
    'LINK_TO' => '#mypopup',
  );?>
  <?=CWD_Popup::BeginEx($arParams);?>
  Здесь содержимое окна.
  <?=CWD_Popup::EndEx();?>
<?endif?>
```

Краткое пояснение данного кода: инструкция `CModule::IncludeModule('webdebug.popup')` подключает системные файлы модуля (PHP-файлы), как и любой другой модуль в 1С-Битрикс. Инструкция `CWD_Popup::Init(true)` подключает на страницу клиентские файлы всплывающего окна: стили и скрипты (CSS и JS-файлы). Переменная `$arParams` – это массив с параметрами компонента. В примере указан, конечно, далеко не весь набор параметров, а лишь минимально необходимый. Инструкция `CWD_Popup::BeginEx($arParams)` начинает HTML-вывод (разметку) всплывающего окна, а инструкция `CWD_Popup::EndEx()` заканчивает его.

Расположение компонента внутри других

Если компонент всплывающего окна или его API-вызов находятся внутри другого компонента, работающего в режиме кеширования, то всплывающее окно не будет корректно работать. Причина заключается в том, что в этом случае вызов компонента не выполняется, а просто выводится все то же, что компонент выводил в последний раз, инструкция `CWD_Popup::Init(true)`, подключающая необходимые файлы, не срабатывает. В результате содержимое всплывающего окна видно сразу после загрузки страницы, всплывающее окно и кнопка его открытия не работают.

Решение очевидно и заключается в ручном прописывании указанного кода в части сайта, которая не кешируется. Например, если всплывающее окно расположено в шаблоне компонента `bitrix:catalog.element`, который, в свою очередь расположен внутри комплексного компонента `bitrix:catalog`, то лучше всего прописать этот код в файле `element.php` перед подключением

компонента `bitrix:catalog.element`. Это связано в том, что в данном случае подключаемые CSS и JS-файлы не будут загружаться на страницах, где нет всплывающих окон.

В случае, если это не комплексный компонент, а, например, простой `bitrix:catalog.element`, то указанный код легче всего вставить в шаблон сайта в файл `header.php` (учтите, что если сайт использует несколько шаблонов, то нужно вставить код во все шаблоны, использующие всплывающие окна).

Указанный код также должен содержать подключение модуля всплывающих окон. Поэтому в двух приведенных случаях код для ручного прописывания следующий:

```
<?if(CModule::IncludeModule('webdebug.popup')){CWD_Popup::Init(true);}?>
```

Также указанный код можно разместить в одном файле `/bitrix/php_interface/init.php`, но с осторожностью: во-первых, если допустить синтаксическую ошибку PHP в этом файле, сайт полностью перестанет работать, во-вторых, этот файл подключается на всех без исключения страницах сайта, использующих API 1С-Битрикс, в т.ч. на страницах административного раздела, и на файлах, загружаемых с помощью AJAX.

Примечание: обновление 1.0.12 решает данную проблему способом автоматического подключения указанных клиентских файлов для всех страниц сайта. В то же время, использование данной опции нежелательно, т.к. она подключает необходимые файлы на всех страницах сайта, даже на тех где это может не требоваться, из-за чего скорость загрузки может немного увеличиться.

Параметры всплывающих окон

Название параметра	Код	Описание
Параметры содержимого окна		
Показывать включаемую область		Способ выбора файла с содержимым окна. Подробное описание смотрите в разделе «Установка статического всплывающего окна».
Суффикс имени файла включаемой области (только при выбранной опции «Показывать включаемую область» в значении «из файла» и «из раздела»)		
Рекурсивное подключение включаемых областей разделов (только при выбранной опции «Показывать включаемую область» в значении «из раздела»)		Позволяет загружать содержимое всплывающего окна из соответствующего включаемого файла родительского раздела, если в просматриваемом разделе нет своего созданного файла.
Шаблон области по умолчанию		Шаблон содержимого окна, по умолчанию устанавливаемый при

Название параметра	Код	Описание
		первом редактировании.
Параметры всплывающего окна		
ID модального окна (должно быть уникально в пределах страницы)	POPUP_ID	Идентификатор всплывающего окна. Может содержать только латинские буквы, цифры, и символ подчеркивания. Начинаться должен с латинской буквы или символа подчеркивания. На просматриваемой странице не должно быть одновременно всплывающих окон с одинаковым ID.
Заголовок модального окна	POPUP_NAME	Текстовый заголовок всплывающего окна.
Ширина модального окна (в пикселях)	POPUP_WIDTH	Ширина всплывающего окна в пикселях.
Наличие кнопки закрытия	POPUP_CLOSE	Опция позволяет отменить возможность простого закрытия окна как по клику на крестик, так и по клику на затемненный фон, так и по нажатию на Esc. В таком случае закрытие окна необходимо выполнять при нужном событии программным путем.
Текст кнопки закрытия	POPUP_CLOSE_TITLE	Опция задает текст кнопки закрытия окна. По умолчанию это крестик (×).
Перемещать окно в BODY	POPUP_APPEND_TO_BODY	Опция позволяет переместить окно в тело страницы, благодаря чему оно всегда правильно позиционируется. Без включенной опции оно находится в той части сайта, откуда компонент вызван, и позиционируется относительно ближайшего родителя со стилем position:relative. При включенной опции после загрузки страницы всплывающее окно незаметно перемещается в тело страницы (body), при этом если в контенте окна находятся скрипты, они выполняются два раза.
Скрывать окно	POPUP_DISPLAY_NONE	Опция предназначена для первоначального скрытия окна,

Название параметра	Код	Описание
		устанавливая <code>display:none</code> вместо <code>visibility:hidden</code> (хотя последнее не отменяется). Это необходимо, если во всплывающем окне расположено, например, Flash-видео, которое без данной опции показывается даже при невидимом окне. Также опция применяет <code>display:none</code> и при закрытии окна.
Эффект анимации	POPUP_ANIMATION	Эффект анимации. Может быть либо «fadeAndPop», либо «fade», либо «none».
JS-функция при создании всплывающего окна	POPUP_CALLBACK_INIT	Имя функции-коллбэка, которая будет выполняться в момент создания всплывающего окна (задолго до показа). Может применяться, например, для модификации окна, или для получения информации о том, что окно уже открывалось ранее.
JS-функция при первом открытии всплывающего окна	POPUP_CALLBACK_OPEN	Имя функции-коллбэка, которая будет выполняться непосредственно перед первым открытием данного окна. Может применяться, например, для AJAX-загрузки содержимого один раз.
JS-функция при каждом открытии всплывающего окна	POPUP_CALLBACK_SHOW	Имя функции-коллбэка, которая будет выполняться непосредственно перед каждым открытием данного окна. Может применяться, например, для AJAX-загрузки содержимого множество раз.
JS-функция при закрытии всплывающего окна	POPUP_CALLBACK_CLOSE	Имя функции-коллбэка, которая будет выполняться непосредственно перед закрытием данного окна. Может применяться, например, для отмены закрытия или для сохранения информации о том, что данное окно открывалось.
CSS-классы всплывающего окна	POPUP_CLASSES	Массив (набор) CSS-классов, которые будут применены ко всплывающему окну. По умолчанию доступны классы от <code>wd_popup_style_01</code> до <code>wd_popup_style_07</code> , которые задают визуальное оформление

Название параметра	Код	Описание
		окон.
Показывать ссылку открытия всплывающего окна	POPUP_LINK_SHOW	При отмеченной опции компонент покажет ссылку для открытия всплывающего окна, в ином случае
jQuery-селектор для открываемой ссылки (только при неотмеченной опции «Показывать ссылку открытия всплывающего окна»)	POPUP_LINK_TO	Позволяет вручную задать, по щелчку на каких элементах будет открываться всплывающее окно. Например, можно задать открытие окна по всем ссылкам, имеющим класс «open»: нужно задать «a.open» (без кавычек).
Текст ссылки (только при отмеченной опции «Показывать ссылку открытия всплывающего окна»)	POPUP_LINK_TEXT	Задаёт текст ссылки, которую создает сам компонент.
Фиксированное окно	POPUP_FIXED	Опция позволяет создать всплывающее окно фиксированным (position:fixed), благодаря чему оно будет видно всегда, вне зависимости от прокрутки страницы. Используйте с осторожностью при большом объеме содержимого во всплывающем окне, т.к. в таком случае нижняя часть окна может выходить за границы окна (особенно у пользователей с небольшим разрешением экрана).
Выравнивать по вертикали по центру	POPUP_MIDDLE	Опция позволяет располагать всплывающее окно по центру страницы. Для фиксированных окон данная опция действует также при изменении размеров браузера.
Автооткрытие окошка	POPUP_AUTOOPEN	Опция включает автоматическое открытие всплывающего окна через количество миллисекунд, указанное в следующем параметре.
Задержка перед открытием, мс (только при отмеченной опции «Автооткрытие окошка»)	POPUP_AUTOOPEN_DELAY	Параметр задает время (в миллисекундах), по прошествии которого автоматически открывается всплывающее окно. Отсчет времени начинается после загрузки страницы.

Название параметра	Код	Описание
Показывать только один раз для пользователя	POPUP_AUTOOPEN_ONCE	Опция реализует однократный (в течение указанного в другом параметре времени) автоматический показ всплывающего окна для посетителя сайта.
Срок хранения информации об открытии окна пользователем (дней)	POPUP_AUTOOPEN_TERM	Время, в течение которого пользователю не будет показываться автоматически открываемое окно. Например, при значении «10», пользователь будет видеть окно 1 раз в 10 дней.
Учитывать раздел сайта	POPUP_AUTOOPEN_PATH	Данная опция позволяет задавать различные настройки для разных разделов сайта. Если этого не требуется – не отмечайте данную опцию.
Скрывать содержимое всплывающего окна в исходном коде страницы, если оно не будет показано в очередной раз	POPUP_AUTOOPEN_FULLHIDE	Опция позволяет скрыть из кода страницы контент окна, если оно не будет показано пользователю. Т.е. код окна не будет исполнен и не будет показан. Помогает, например, чтобы ускорить загрузку страницы без всплывающего окна.
Скрытая ссылка	POPUP_LINK_HIDDEN	При отмеченной опции ссылка, создаваемая компонентом, является скрытой (с заданным стилем display:none); всплывающее окно с такой ссылкой можно открывать либо автоматически, либо вручную.

Для динамических всплывающих окон код параметра не содержит «POPUP_», т.е. в этом случае параметры это «ID», «NAME», «WIDTH», «CLOSE» и т.д.

Описание API модуля

В данном разделе будет описан исключительно JavaScript API, т.к. API PHP предназначено для использования чисто во вспомогательных целях.

function WD_Popup_Init() {}

Функция инициализирует механизм всплывающих окон: определяются элементы, при щелчке на которые открываются всплывающие окна, задаются параметры окон, назначаются действия при

щелчке, выполняется событие при создании окон, запускается отсчет на автоматическое открытие всплывающих окон.

Функция выполняется автоматически, но может быть полезна, когда страница со всплывающими окнами загружается в режиме AJAX, т.к. в этом случае функция автоматически не запускается.

Внимание! До версии 1.0.23 функция имела название «WD_Popups_Init».

function WD_Popup_AJAX(Container, URL, FormData, Callback) {}

Функция выполняет AJAX-запрос. Данная функция является оберткой для функции \$.ajax(), с заданным методом запроса POST.

Параметры:

- **Container** – jQuery-объект, содержащий контейнер, в который будет записан результат,
- **URL** – URL (адрес) страницы, к которой выполняется запрос,
- **FormData** – сериализованные данные формы, например, \$("#myform").serialize(),
- **Callback** – функция, вызываемая при выполнении запроса. Задается непосредственно функцией (не именем). Функция получает следующие параметры: res, Container, URL, FormData, где res – ответ, полученный от сервера.

function WD_Popup_GetContentObject(ID) {}

Функция получает jQuery-объект элемента, содержащего контент всплывающего окна с идентификатором ID.

function WD_Popup_Close(ID) {}

Функция закрывает одно или несколько окон. Если ID равен false, будут закрыты все окна, иначе – только окно с указанным идентификатором.

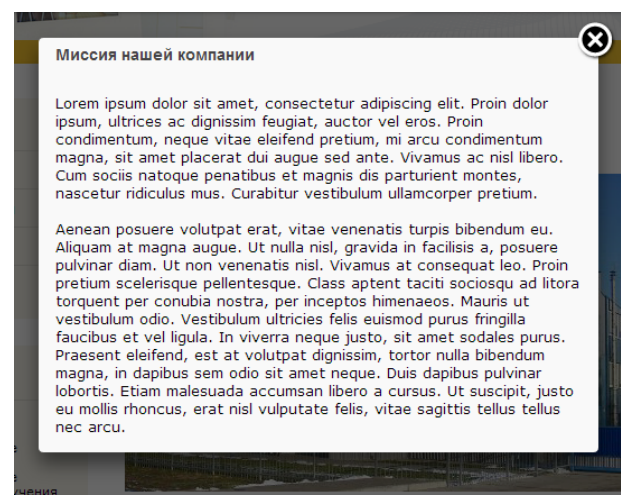
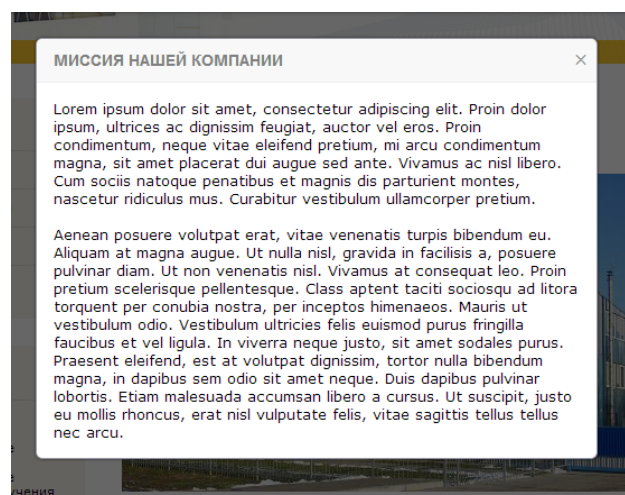
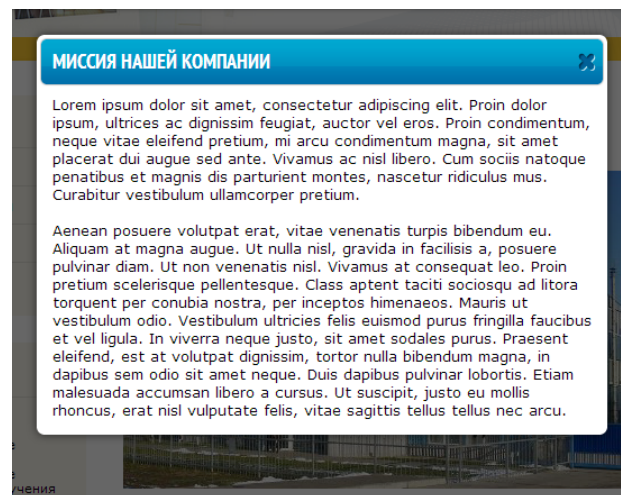
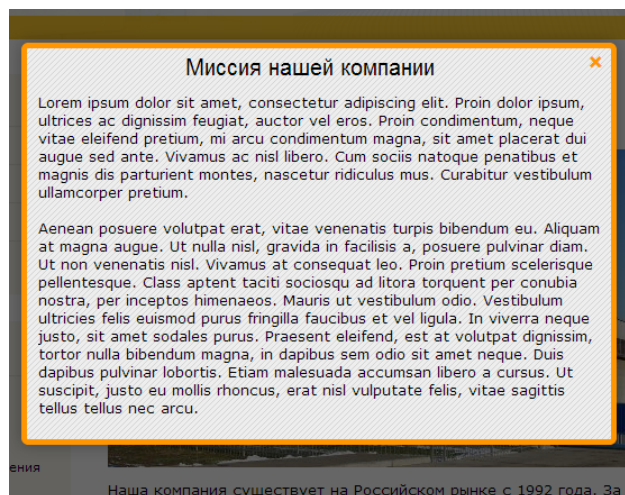
function WD_Popup_Open(ID) {}

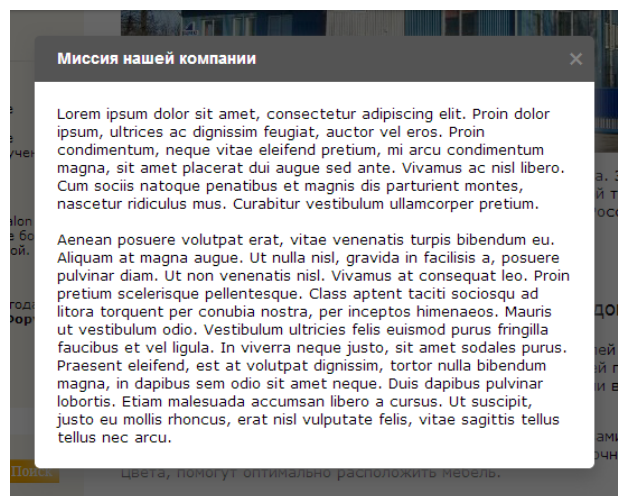
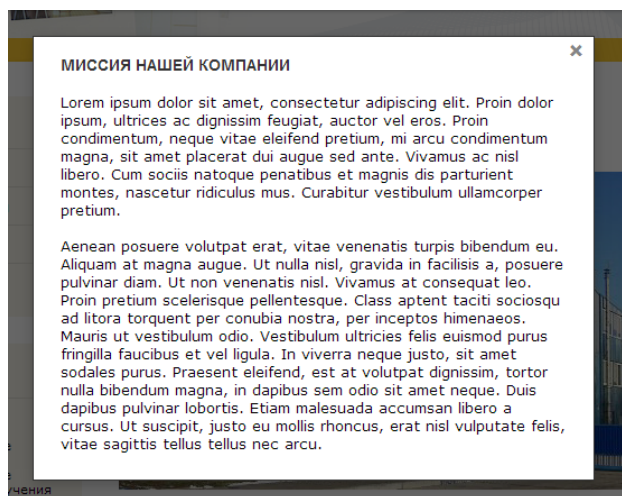
Функция программно открывает окно, эмулируя клик по соответствующей ссылке. Не будет работать, если установлена опция «Показывать ссылку открытия всплывающего окна» - в таком случае нужно для своей ссылки эмулировать клик самостоятельно.

Встроенные схемы оформления

Модуль имеет несколько готовых цветовых схем оформления. Цветовые схемы определяются в соответствии с разработанными стилями всплывающих окон, которые размещаются в файле `/bitrix/themes/.default/webdebug.popup.css`. В указанном файле в первую очередь указаны стили, реализующие основные свойства, не имеющие отношения к дизайну (размер, расположение, поведение, видимость), а затем – готовые цветовые схемы оформления: `wd_popup_style_01`, `wd_popup_style_02`, `wd_popup_style_03`, `wd_popup_style_04`, `wd_popup_style_05` (используется по умолчанию), `wd_popup_style_06`.

Использование продолжающейся нумерации (напр., «wd_popup_style_20») крайне не рекомендуется, т.к. с развитием модуля в него будут добавлены новые цветовые схемы оформления, которые и будут использовать данную нумерацию, из-за чего созданные стили будут конфликтовать с новыми.





Такой механизм определяет простой путь стилизации всплывающих окон. Порядок стилизации следующий:

- 1) скопировать один из вариантов цветовых схем (три строчки) в любой из используемых на сайте файлов стилей (см. далее по тексту), и заменить цифру в обозначении (либо полностью поменять обозначение), напр., «`my_popup_style`» (можно использовать только латинские буквы, цифры и символ подчеркивания, причем обозначение может начинаться только с латинской буквы),
- 2) в скопированном коде поменять стили в соответствии с необходимыми настройками,
- 3) в настройках компонента указать используемое обозначение, напр., «`my_popup_style`».

Собственный стиль оформления лучше разместить либо в файле стилей шаблона сайта (`template_styles.css`), либо в файле стилей сайта (`styles.css`), либо копировать шаблон компонента всплывающего окна (`webdebug:popup_window`) и в шаблоне создать файл `style.css`, куда и прописывать новые стили.

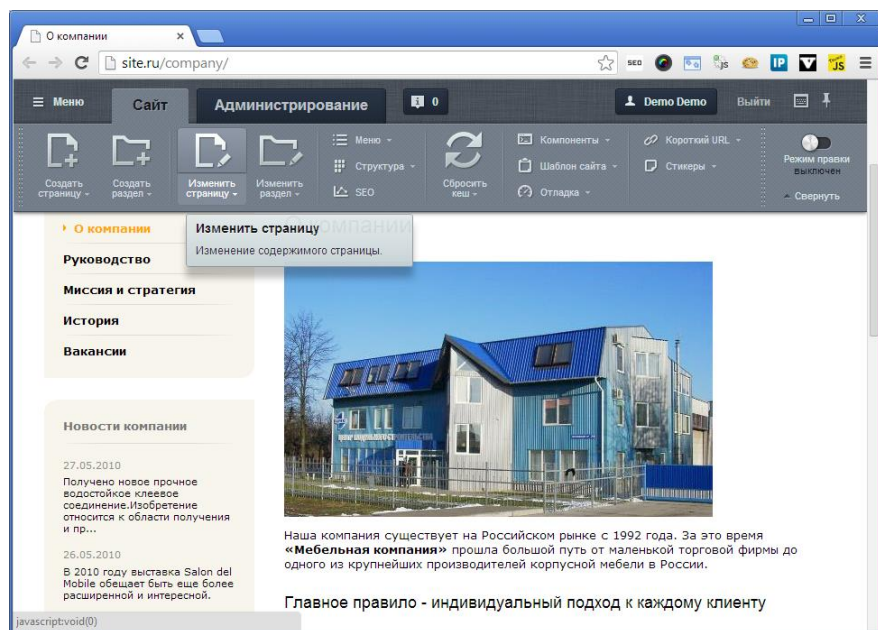
Примеры установки и настройки

В рамках данного руководства будем устанавливать компонент всплывающих окон на стандартное готовое решение от 1С-Битрикс «Корпоративный сайт производственной компании».

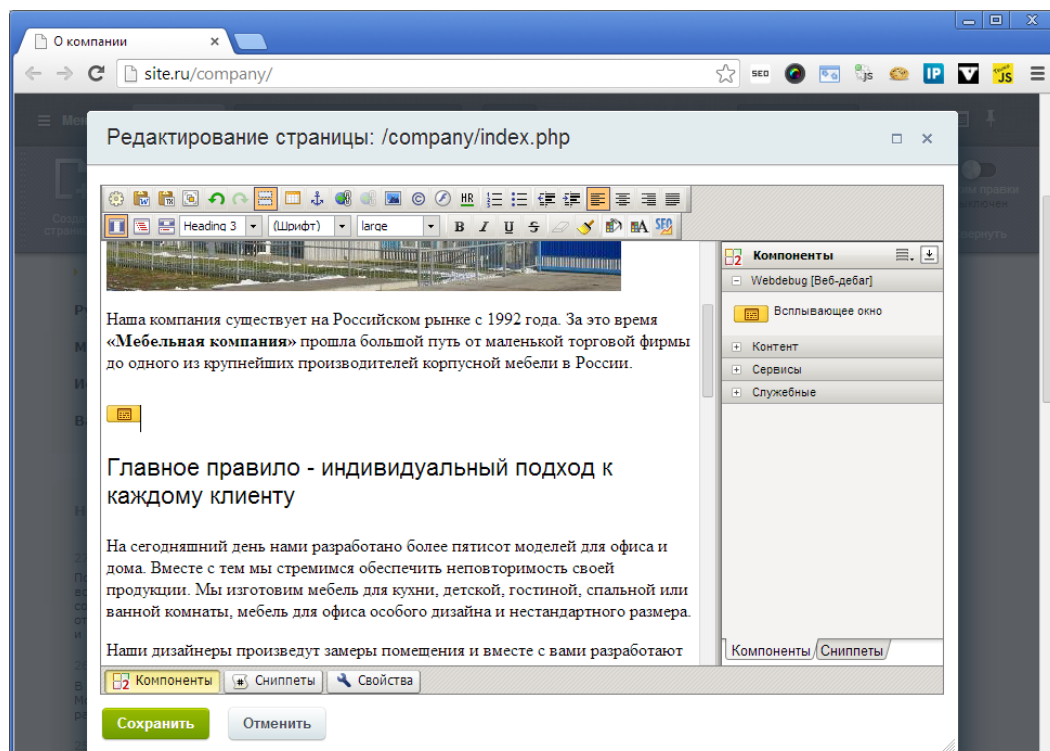
Простейший пример установки всплывающего окна.

Простейший пример установим в разделе «Компания»: сделаем всплывающее окно, которое будет содержать текст миссии компании. Ссылку открытия окна расположим сразу после первого абзаца текста, следующего за фотографией.

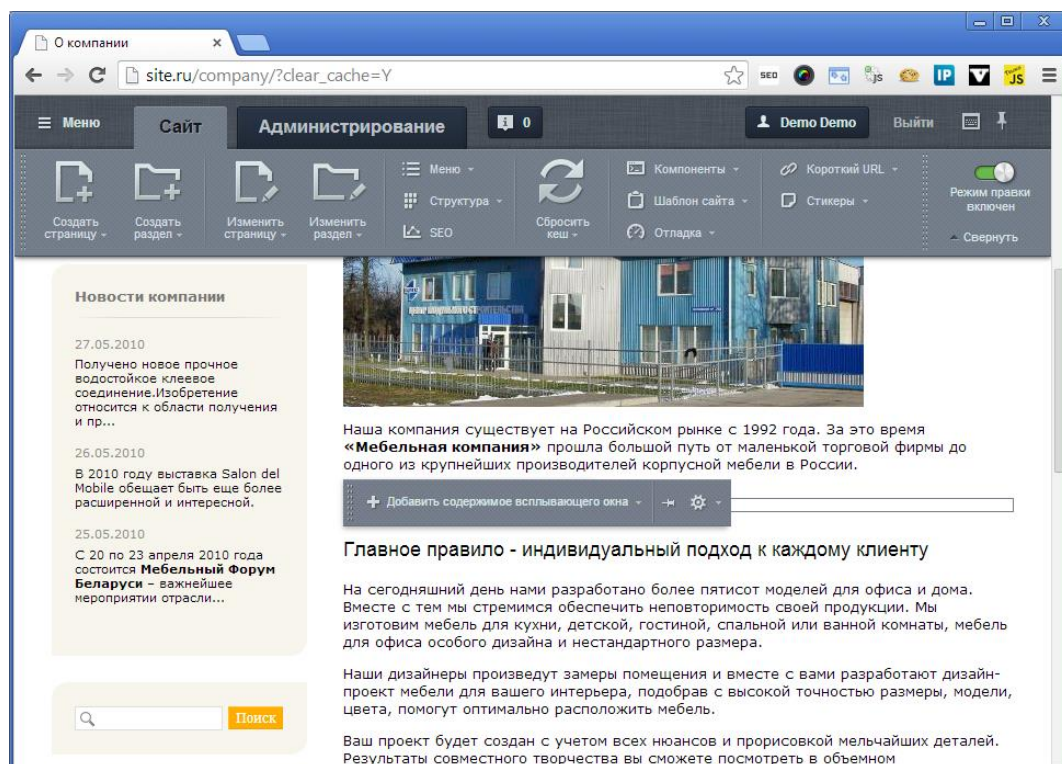
Для начала переходим на страницу «Компания» и на панели инструментов нажимаем кнопку «**Изменить страницу**»:



В открывшемся визуальном редакторе добавляем пустую строку в месте, где хотим разместить компонент. Из правой панели устанавливаем компонент «Всплывающее окно», которое находится в секции «Webdebug [Веб-дебаг]». Если последующий заголовок «налез» на строку с компонентом, переносим его на новую строку. Сохраняем нажатием зеленой кнопки.



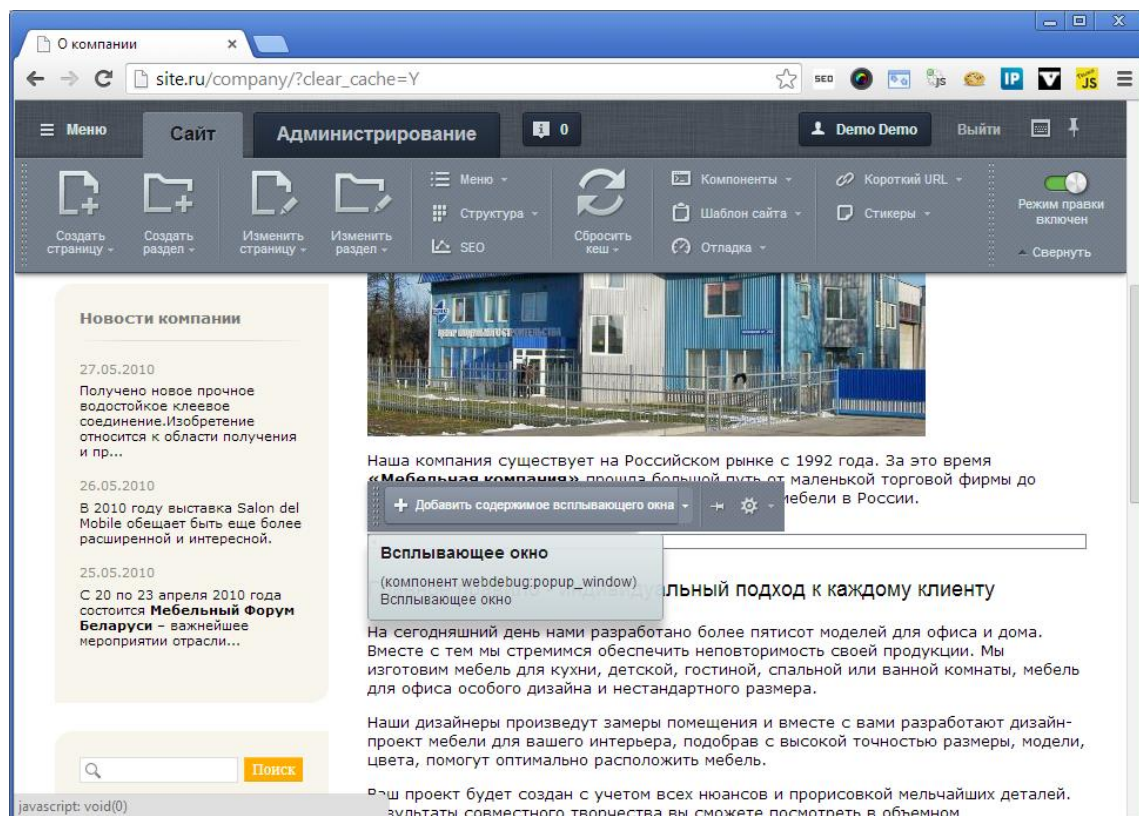
Переходим в режим правки. Наводим курсором в то место, где должен быть компонент (подсветится узкая полоска). Нажимаем кнопку с картинкой шестеренки – настройки.



Настраиваем компонент:

Параметр	Значение
Показывать включаемую область	для страницы
Суффикс имени файла включаемой области	mission
ID модального окна	mission
Заголовок модального окна	Миссия нашей компании
Ширина модального окна (в пикселях)	500
Наличие кнопки закрытия	Да
Перемещать окно в BODY	Да
Скрывать окно	Нет
CSS-классы всплывающего окна	--- оставить пустым --- или wd_popup_style_05
Показывать ссылку открытия всплывающего окна	Да
Текст ссылки	Открыть текст о миссии компании
Автооткрытие окошка	Нет
Задержка перед открытием, мс	--- не важно ---
Скрытая ссылка	Нет

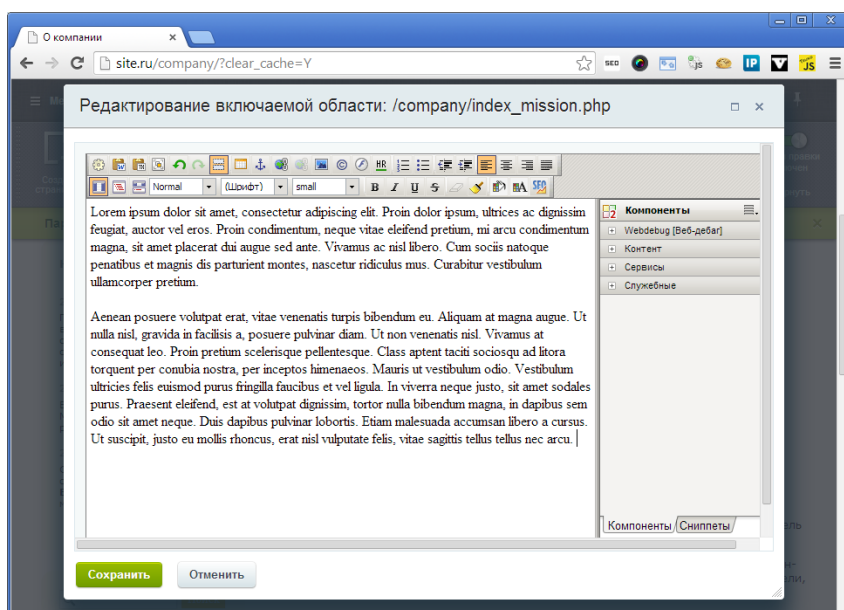
Теперь, когда все параметры выставлены, осталось создать содержимое окна. Для этого в режиме правки опять наводим курсором на узкую полоску области с компонентом. Появляется всплывающая контекстная панель, на которой нужно нажать «**Добавить содержимое всплывающего окна**». В дальнейшем, когда файл с содержимым уже будет создан, здесь будет кнопка «Изменить содержимое всплывающего окна». После нажатия кнопки открывается визуальный редактор.

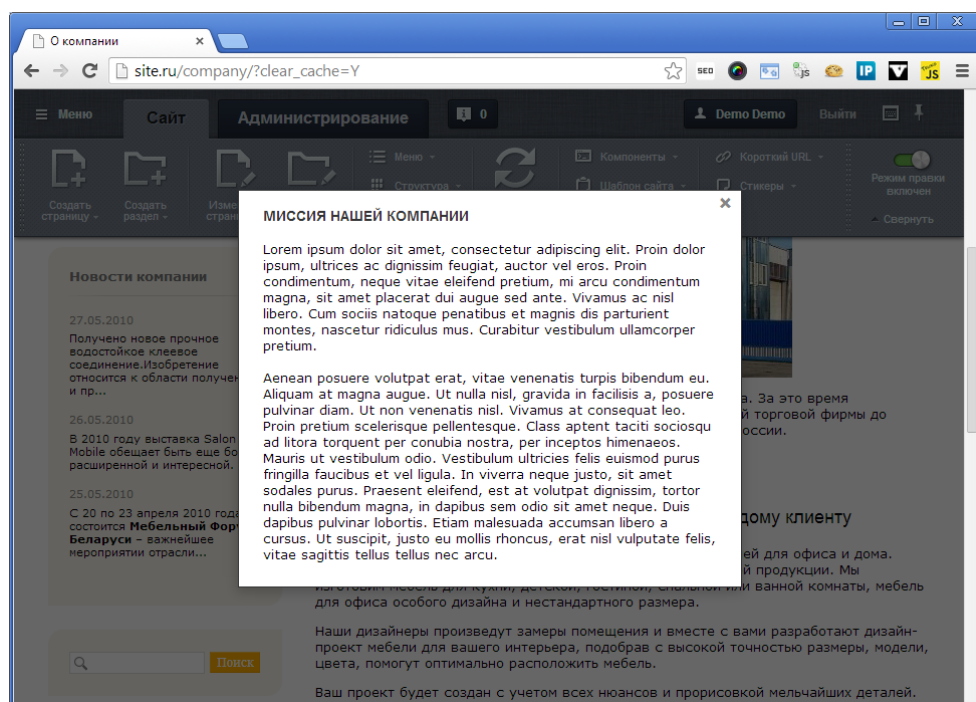


В заголовке визуального редактора файл `/company/index_mission.php` указан как `/company/index_mission.php`, где `index` – это часть, взятая из текущей страницы (настоящая текущая страница `/company/index.php` – `index.php` не отображается, но подразумевается), а `mission` – это параметр «Суффикс имени файла включаемой области». Т.о., если в разделе создать файл `/company/about.php`, то содержимое всплывающего окна для типа «для страницы» будет иметь вид `about_mission.php`.

В визуальном редакторе нужно вставить то, что должно быть во всплывающем окне. В нашем случае это просто два абзаца текста. После редактирования нужно нажать кнопку «Сохранить.»

В результате на странице появляется ссылка «Открыть», при нажатии на которую открывается всплывающее окно с текстом:





Все остальные примеры подключения статических всплывающих окон аналогичны, за исключением некоторых особенностей в настройке. Поэтому данный пример наиболее важен для понимания.

Абсолютно аналогично размещаются и статические картинки, таблицы, карта сайта, новости, статьи и т.п.

Всплывающее окно с видео из YouTube.com

В целом процесс создания всплывающего окна с видео из YouTube абсолютно аналогичен. Разница заключается лишь в том, что в настройках компонента необходимо отметить опцию «Скрывать окно», указать правильную ширину окна с учетом отступа, а в визуальном редакторе вставлять не текст, а HTML-код для вставки видео из YouTube.com.

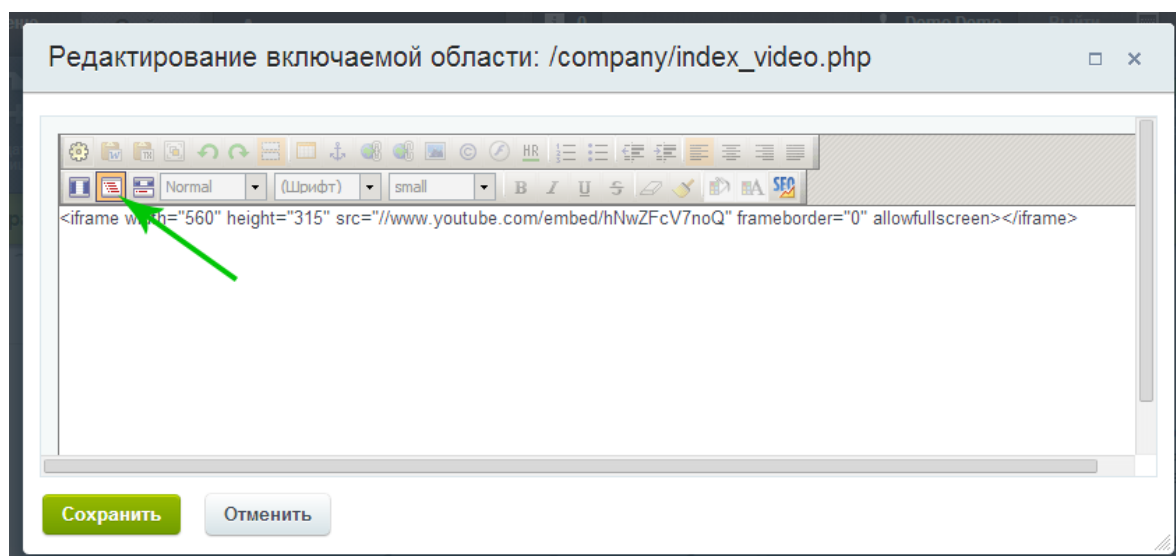
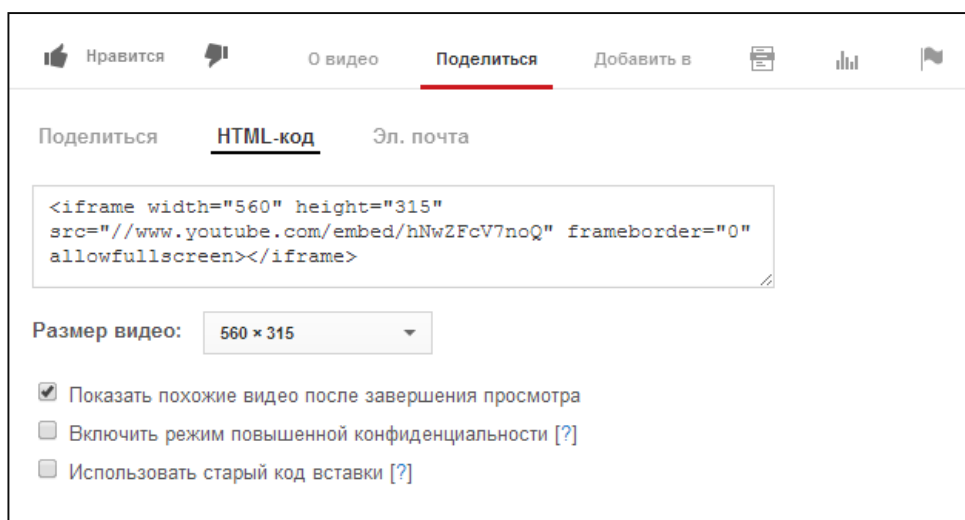
В данном примере настройки следующие:

Параметр	Значение
Показывать включаемую область	для страницы
Суффикс имени файла включаемой области	video
ID модального окна	Video
Заголовок модального окна	Видео о компании
Ширина модального окна (в пикселях)	606
Наличие кнопки закрытия	Да
Перемещать окно в BODY	Да
Скрывать окно	Да
CSS-классы всплывающего окна	--- оставить пустым --- или wd_popup_style_05

Показывать ссылку открытия всплывающего окна	Да
Текст ссылки	Открыть текст о миссии компании
Автооткрытие окошка	Нет
Задержка перед открытием, мс	--- не важно ---
Скрытая ссылка	Нет

Видео с разных сайтов и сервисом вставляются разными способами. Наиболее просто вставить видео именно из YouTube.com.

Для получения кода вставки перейдите на страницу видео в YouTube.com, под блоком с видео нажмите на кнопку «Поделиться», затем – на кнопку «HTML-код». Откроется текстовое поле, его содержимое нужно скопировать, и затем вставить в визуальном редакторе в режиме HTML-кода (переход в HTML-режим – вторая кнопка слева на нижней панели). После этого – сохранить.0



Имейте ввиду, что задаваемая ширина окна измеряется границами внешнего блока окна. Поэтому, область содержимого окна меньше указываемой ширины на размер отступов слева и справа. Отступы задаются в стилях CSS.

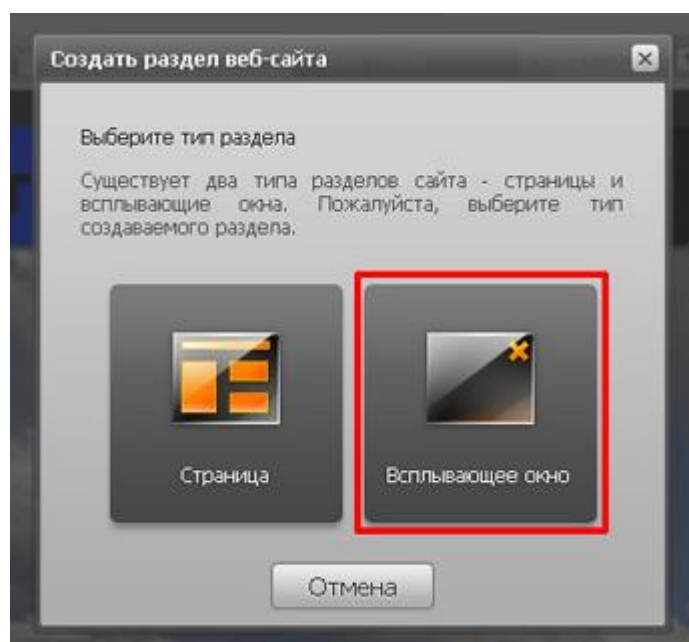
После сохранения получаем ссылку «Видео о компании», при нажатии на которую открывается всплывающее окно:



При закрытии окна происходит присвоение стиля `display:none`, благодаря чему видео принудительно останавливается.

Пример создания собственного стиля оформления

Предположим, перед нами встала задача настроить всплывающие окна в следующем стиле:



Открываем файл `/bitrix/themes/.default/webdebug.popup.css`, копируем три строчки с правилами оформления для стиля `wd_popup_style_01`. Удаляем все CSS-свойства:

```
.wd_popup_style_01 {}  
    .wd_popup_style_01 .wd_popup_title {}  
    .wd_popup_style_01 .wd_popup_close {}  
    .wd_popup_style_01 .wd_popup_close:hover {}
```

В указанном CSS-коде все достаточно понятно - сверху вниз: стиль самого окна, стиль заголовка, стиль кнопки закрытия, и кнопка закрытия при наведении курсора. К указанным стилям можно только добавить:

```
.wd_popup_style_01 .wd_popup_inner {}
```

Этот стиль определяет внешний вид области с содержимым.

Модуль автоматически присваивает кнопке закрытия HTML-крестик как наиболее универсальный элемент. Обойти эту автоматику можно, опять-таки, CSS-правилами.

После этого приступаем к созданию своего собственного стиля. Вся работа заключается в настройке CSS-стилей. Для начала создадим градиент окна. Помочь может специальный онлайн-сервис <http://www.colorzilla.com/gradient-editor/> (необходимо только задать начальный и конечный цвета).

После этого настраиваем рамки (`border`), отступы (`padding`, `margin`), скругления (`border-radius`) затем приводим в порядок заголовки окна (`color`, `font`, `height`, `text-shadow`), затем – кнопку закрытия (картинка взята из случайно найденных с помощью Гугл.Картинок), затем – фон области контента.

Затем названия были заменены в соответствии с желаемым названием новой схемы оформления: «`my_popup_style`».

В результате настройки стилей получен следующий набор CSS-стилей (он выглядит некрасиво из-за большого количества стилей, добавляемых для реализации кроссбраузерных градиентов):

```
.my_popup_style {background:#545452; border:1px solid #8b8b8b; padding:0 3px  
3px; -moz-border-radius:3px; -webkit-border-radius:3px; border-radius:3px;}  
    .my_popup_style .wd_popup_title {  
        color:#fff; font:bold 12px/24px "Tahoma","Geneva",sans-serif;  
height:24px; margin:0 -3px; overflow:hidden; padding:0 8px;  
        text-shadow:0 0 1px #ccc;  
        background: #747474;  
        background:  
url(data:image/svg+xml;base64,PD94bWwgdmVyc2lvdj0iMS4wIiA/Pgo8c3ZnIHhtbG5zPSJo  
dHRwOi8vd3d3LnczLm9yZy8yMDAwL3N2ZyIgd2lkdG99IjEwMCUuIGhlaWdodD0iMTAwJSIgdmlld0  
JveD0iMCAwIDEgMSIgcHJlc2Vydmlvc3B1Y3RSYXRpbz0ibm9uZSI+CiAgPGxpbmVhckdyYWRpZW50  
IGlkPSJncmFkLXVjZ2ctZ2VuZXJhdGVkIiBncmFkaWVudFVuZXJhdGVudD0iMTAwJSIgdmlld0  
E9IjAlIiB5MT0iMCAwIDEgMSIgcHJlc2Vydmlvc3B1Y3RSYXRpbz0ibm9uZSI+CiAgICA8c3RvcCBvZmZzZXQ9IjEwMC  
UuIHN0b3AtY29sb3I9IiM0OTQ5NDkiIHN0b3Atb3BhY210eT0iMSIvPgogIDwvbGluZWFyR3JhZGl1  
bnQ+CiAgPHJlY3QgeD0iMCAwIDEgMSIgd2lkdG99IjEwMCUuIGhlaWdodD0iMSIgdmlsbD0idXJsKCNcm  
FkLXVjZ2ctZ2VuZXJhdGVkKSIGLz4KPC9zdmc+);  
        background: -moz-linear-gradient(top, #747474 0%, #494949  
100%);
```



```

background: -webkit-gradient(linear, left top, left bottom,
color-stop(0%,#747474), color-stop(100%,#494949));
background: -webkit-linear-gradient(top, #747474 0%,#494949
100%);
background: -o-linear-gradient(top, #747474 0%,#494949 100%);
background: -ms-linear-gradient(top, #747474 0%,#494949 100%);
background: linear-gradient(to bottom, #747474 0%,#494949
100%);
filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#747474', endColorstr='#494949',GradientType=0 );
}
.my_popup_style .wd_popup_close
{background:url("https://www.ftb.ca.gov/images/common/cancel.png") 0 0 no-
repeat; font-size:0; height:16px; width:16px;} .my_popup_style
.wd_popup_close:hover {}
.my_popup_style .wd_popup_inner {background:#c6c4c5; border-bottom:1px
solid #747273; border-top:1px solid #747273; color:#333; padding:20px 18px; -
moz-border-radius:2px; -webkit-border-radius:2px; border-radius:2px;}

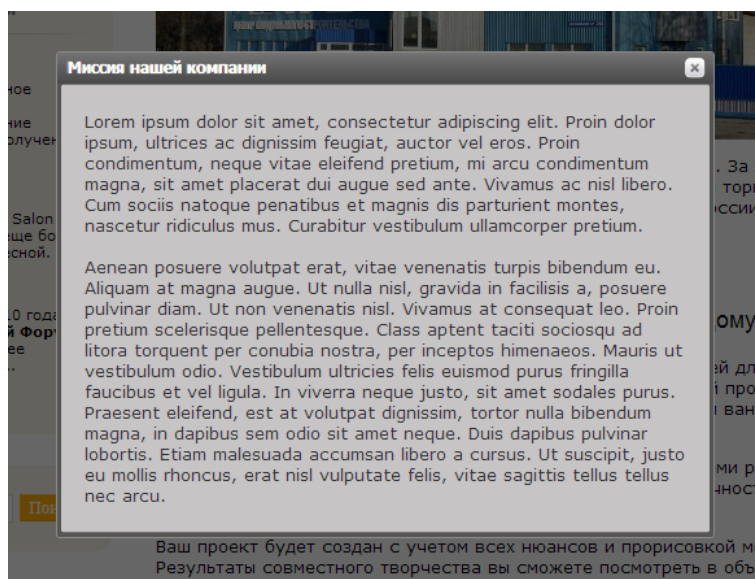
```

Эти CSS-стили копируем в файл стилей шаблона сайта. Т.к. пример мы реализуем на готовом решении 1С-Битрикс «Корпоративный сайт производственной компании», то в нашем случае это файл (название шаблона сайта зависит от цветовой схемы сайта, выбранной при установке):

```
\bitrix\templates\furniture_gray\template_styles.css
```

После копирования стилей в указанный файл осталось лишь задать в настройках компонента (параметр «CSS-классы всплывающего окна») класс «my_popup_style», все другие классы готовых цветовых схем должны быть удалены, чтобы стили не смешались. Вспомогательные классы, которые находятся под Вашим контролем, можно оставить.

В результате проделанной работы мы получили всплывающее окно, весьма похожее на пример. Для большей схожести требуется более тщательная работа со стилями.



Т.о., создание своих стилей всплывающих окон – довольно легко. Легкости также добавляет отсутствие необходимости настраивать все системные CSS-стили, связанные с размером, расположением, поведением и др. Это все делает за нас модуль.

Возможны варианты, когда только с помощью CSS-стилей не получится добиться желаемого результата, это случаи когда в окно необходимо добавить новые HTML-элементы, реализующие новые части окна. В таком случае задача может быть легко решена с помощью JS-события создания окна, при котором есть возможность добавить к окну необходимые элементы, например, с помощью jQuery-функций `append()`, `appendTo()` и многих других.

Простейший пример подключения динамического компонента

Представим, что наша задача – на страницу подробного просмотра товаров добавить кнопку открытия всплывающего окна с описанием товара и ценой.

Определяем, какой используется шаблон. В нашем случае это:

```
/bitrix/templates/furniture_gray/components/bitrix/catalog/.default/bitrix/catalog.element/.default/template.php
```

Сначала добавляем ручной вызов компонента и его инициализацию, о чем было указано ранее (для предотвращения проблем с кешированием). Вставляем код

```
<?if(CModule::IncludeModule('webdebug.popup')){CWD_Popup::Init(true);}?>
```

в файл `/bitrix/templates/furniture_gray/components/bitrix/catalog/.default/element.php`. Эту строчку нужно вставить как можно ближе к началу файла, обязательно – перед вызовом компонента `bitrix:catalog.element`.

Далее, в шаблоне компонента `bitrix:catalog.element` (в указанный ранее файл) прописываем подключение всплывающего окна:

```
<?if(CModule::IncludeModule('webdebug.popup')):?>
  <?CWD_Popup::Init(true);?>
  <input type="button" value="Show text!" id="product_open" />
  <?$arPopup = array(
    'ID' => 'product_popup',
    'NAME' => $arResult['NAME'],
    'WIDTH' => '600',
    'CLOSE' => 'Y',
    'APPEND_TO_BODY' => 'Y',
    'CLASSES' => array('wd_popup_style_05'),
    'LINK TO' => '#product_open',
  );?>
  <?=CWD_Popup::BeginEx($arPopup);?>
  <div class="text">
    <?=$arResult['DETAIL_TEXT']?>
  </div>
  <div class="price">
    <?foreach($arResult["PRICES"] as $code=>$arPrice):?>
      <?if($arPrice["PRINT_VALUE"] > 0):?>
```

```
                <div class="catalog-item-  
price"><span><?=GetMessage('CR_PRICE')?>:</span>  
<?=$arPrice["PRINT_VALUE"]?></div>  
                <?endif;?>  
            <?endforeach;?>  
</div>  
<?=CWD_Popup::EndEx();?>  
<?endif?>
```

Здесь `<input ... />` - это элемент управления, по щелчку на который будет открываться всплывающее окно. Он имеет `ID="product_open"`, и в соответствии с этим в массиве настроек параметр `"LINK_TO"` равен `"#product_open"`.

Все, что заключено между `<?=CWD_Popup::BeginEx($arPopup);?>` и `<?=CWD_Popup::EndEx();?>` будет выведено во всплывающем окне. Все зависит от конкретных задач. В нашем случае выводится только подробное описание и цена.

Возможные проблемы и их решения

При попытке загрузки модуля из 1С-Битрикс: Маркетплейс появляется ошибка «[SYS_ERROR_AB_ACTIVE] Истек период получения обновлений и коммерческой техподдержки».

Проблема заключается в том, что для Вашего сайта закончился срок получения обновлений (который действует 1 год со дня активации лицензии на сайте).

Свяжитесь с нами для решения этой проблемы.

В админке 1С-Битрикс нет раздела «Маркетплейс» («Каталог решений»).

Это говорит о том, что у Вас установлена старая версия 1С-Битрикс. Однако несмотря на это, есть возможность установить модуль в ручном режиме.

Свяжитесь с нами для решения этой проблемы.

После установки в списке компонентов не появился компонент «Всплывающее окно»

Для начала проверьте, что модуль установлен и в списке установленных решений имеет статус «Установлено» (см. рисунок в разделе «Загрузка модуля на сайт»).

Попробуйте очистить кеш на сайте, обновить страницу, используя сочетание клавиш Ctrl-F5.

Если проблема не решена, свяжитесь с нами для её решения.

Компонент показывает ссылку, но при щелчке по ней ничего не происходит.

Возможно, причина в конфликте версий jQuery. Проверьте какая версия подключена: соответствует ли она требованиям модуля (см. в соответствующем разделе данного руководства). Если версия jQuery установлена ниже, чем требуется (1.4.3 и ниже), то решением может являться только замена используемой версии на более новую.

Также причиной данной проблемы могут быть ошибки в коде JavaScript на сайте, не имеющие отношения к модулю. Ошибки можно отследить в консоли Вашего браузера (напр., это доступно в Google Chrome, Mozilla Firefox, Opera, Internet Explorer и др.).

Если проблема не решена, свяжитесь с нами для её решения.

Всплывающее окно расположено криво, или вовсе выходит за границы сайта

Скорее всего, проблема в CSS-стилях. Для начала попробуйте в настройках компонента установить опцию «Перемещать окно в тело страницы». Если стиль оформления был изменен, проверьте его корректность. Также проверьте, не задают ли какие-либо скрипты параметры позиционирования для

всплывающего окна (в этом помогут средства разработчика, имеющиеся во всех основных браузерах).

Если проблема не решена, свяжитесь с нами для её решения.