# Подключение и настройка скрипта-обработчика контента статьи для сворачивания/разворачивания текстовых блоков.

[Настройка и подключение скрипта-обработчика контента статьи для сворачивания/разворачивания текстовых блоков. 1](#_Toc347426385)

[Условия 1](#_Toc347426386)

[Реализация 1](#_Toc347426387)

[Создание специфического суффикса для контейнера страницы 1](#_Toc347426388)

[Расположение целевого файла 2](#_Toc347426389)

[Добавление суффикса имени класса 2](#_Toc347426390)

[Соблюдение минимальных требований к структуре добавляемой статьи 2](#_Toc347426391)

[Принцип работы скрипта 3](#_Toc347426392)

[Настройка jQuery-обработчика и его подключение 4](#_Toc347426393)

[Напутствие 4](#_Toc347426394)

## Условия

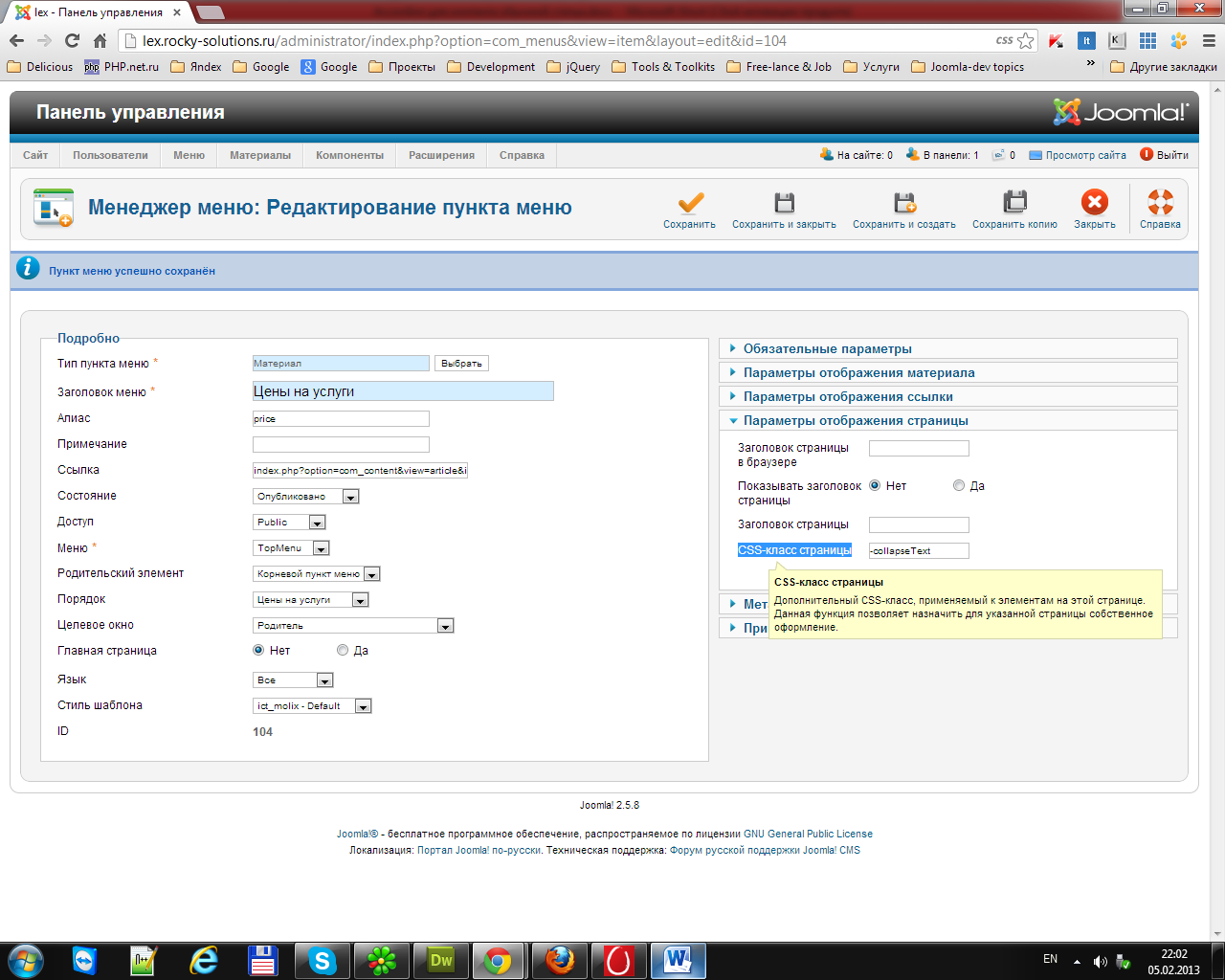
Для реализации функции сворачивания/разворачивания текста внутри блока текста, добавленного в качестве обычной статьи в Joomla!, необходимо выполнение следующих условий:

1. [Создание специфического суффикса для контейнера страницы.](#_Создание_специфического_суффикса)
2. [Соблюдение минимальных требований к структуре добавляемой статьи.](#_Соблюдение_минимальных_требований)
3. [Настройка jQuery-обработчика и его подключение.](#_Настройка_jQuery-обработчика_и)

## Реализация

### Создание специфического суффикса для контейнера страницы

Выполнение данного условия позволяет избирательно воздействовать на таргет-объекты. Создание суффикса для загружаемого модуля — стандартная опция в Joomla! Чтобы назначить суффикс странице, зайдите в раздел редактирования загружающей её ссылки и установите параметр «CSS-класс страницы» в секции «**Параметры отображения страницы**»:

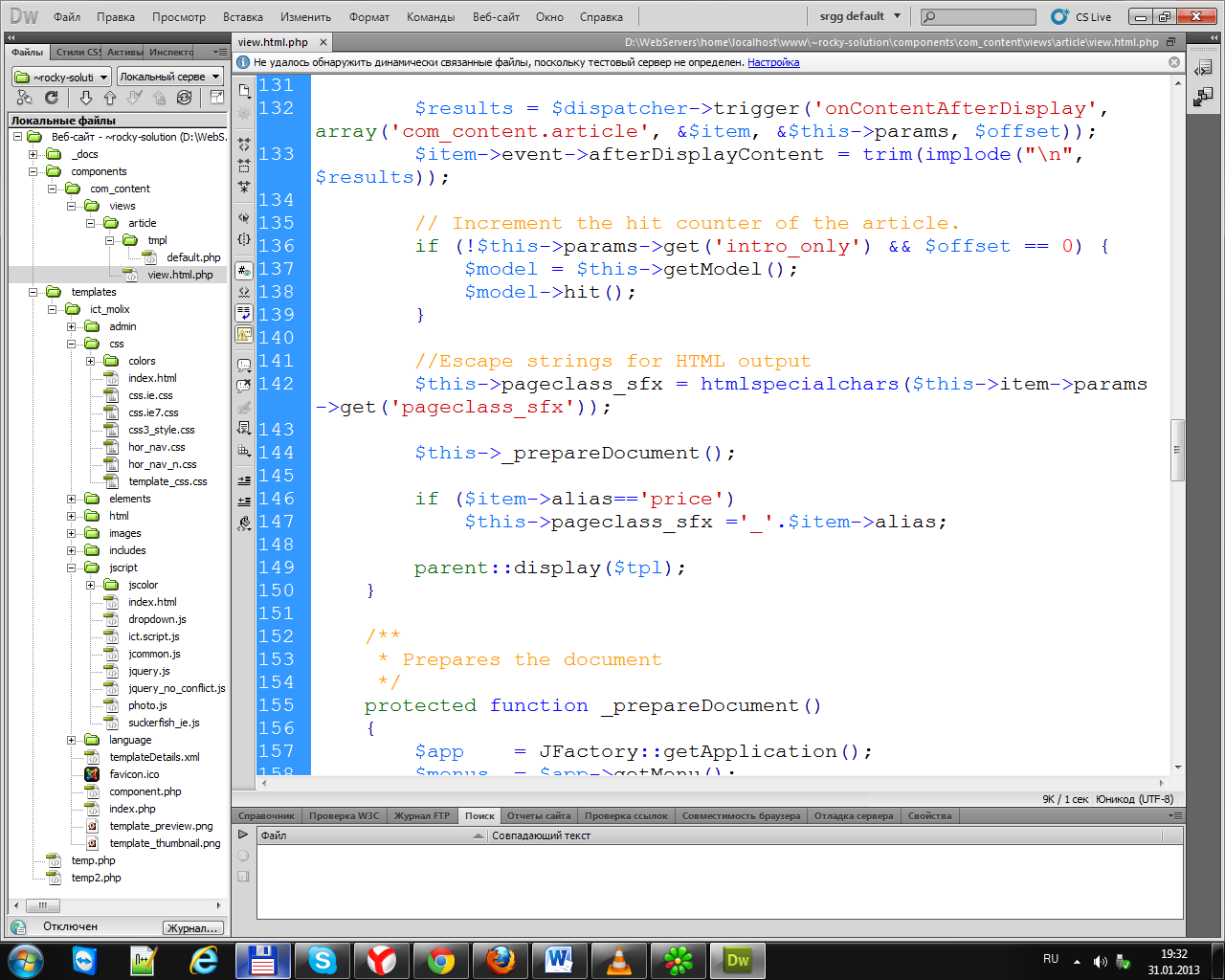


К сожалению, данная процедура не будет доступна, если статья загружается не по ссылке из меню (такое тоже бывает). В этом случае суффикс класса для блока-wrapper’а можно добавить посредством модификации метода ContentViewArticle:: display().

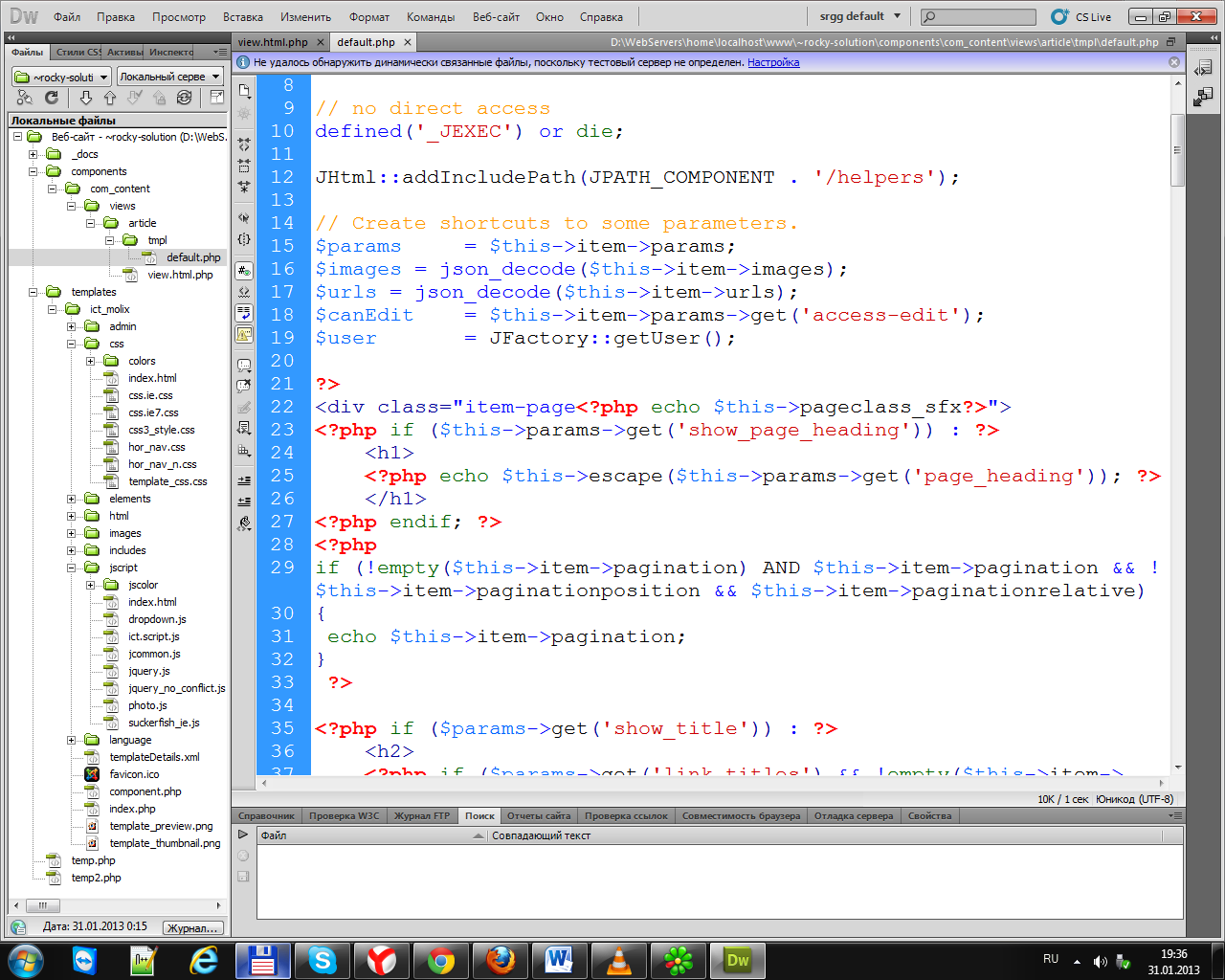
#### Расположение целевого файла

/components/com\_content/views/article/view.html.php

#### Добавление суффикса имени класса



Здесь выполняется проверка алиаса загружаемой страницы (вы можете также использовать другие доступные в URL параметры, например, — id статьи), и суффиксу присваивается данное значение (алиас указывается в панели управления для п. меню, загружающего данную статью). Далее этот суффикс добавляется к имени класса блока, в котором размещается контент статьи:



В результате данному блоку будет присвоено специфическое имя класса, что позволит воздействовать на объекты только на указанных разработчиком страницах (иначе говоря — только для выбранных статей).

### Соблюдение минимальных требований к структуре добавляемой статьи

Обрабатывающий скрипт по своему действию напоминает стандартный jQuery-плагин accordion, однако последний требует соблюдения жёсткой структуры расположения контента, что маловероятно при сохранении текста в виде обычной статьи Joomla!. Разработанный [srgg67@gmail.com](mailto:srgg67@gmail.com), [srgg6701](skype:srgg6701) скрипт, в отличие от него:

1. Не требует обязательного вложения всех элементов текста в единый блок;
2. Позволяет через свой API назначать в качестве trigger-объектов заголовки любого уровня.

Единственными условиями его работы являются:

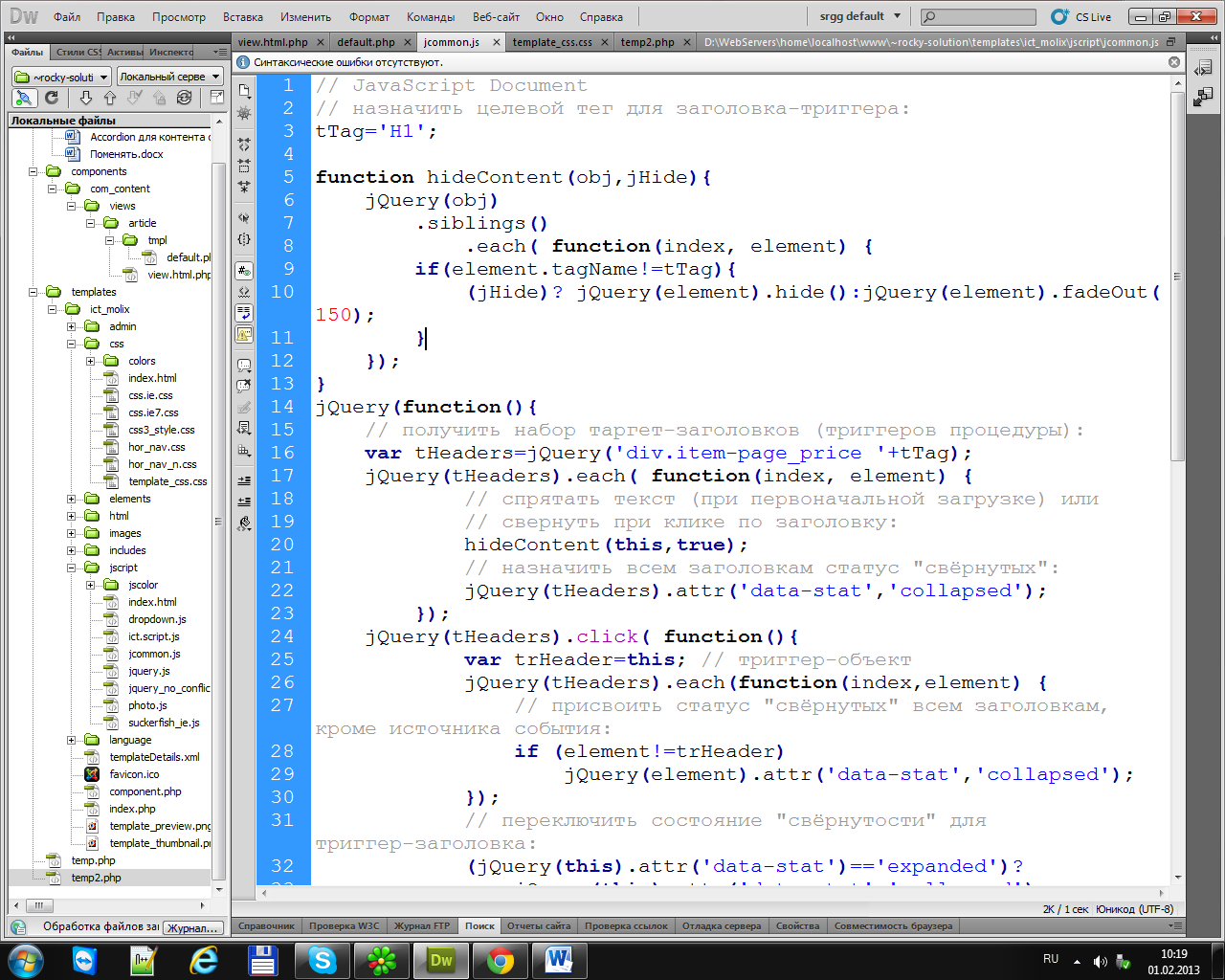
* Всем trigger-объектам должен быть назначен заголовок одного и того же уровня
* Разработчик должен определить HTML-элементы, которые не должны подпадать под воздействие скрипта. По умолчанию первым назначается тег H2, вторыми являются заголовки H1.

### Принцип работы скрипта

Скрипт работает по простому принципу:

#### При начальной загрузке страницы —

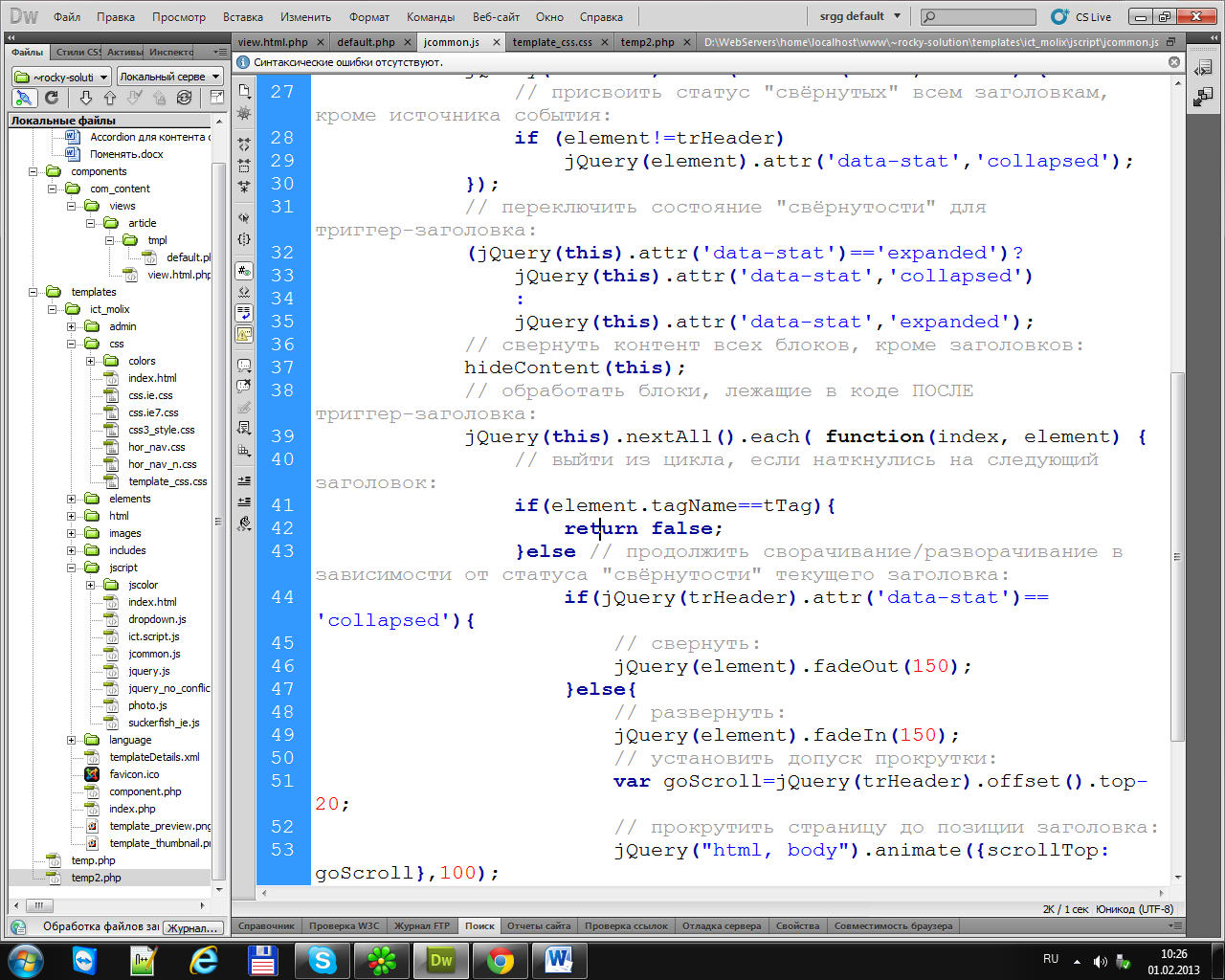
1. сворачивает все текстовые блоки, находящиеся между заголовками, тег которых выбран в качестве целевого для триггер-элемента. Данный тег сохраняется в глобальной переменной, что позволяет получить к нему доступ из любой функции скрипта:



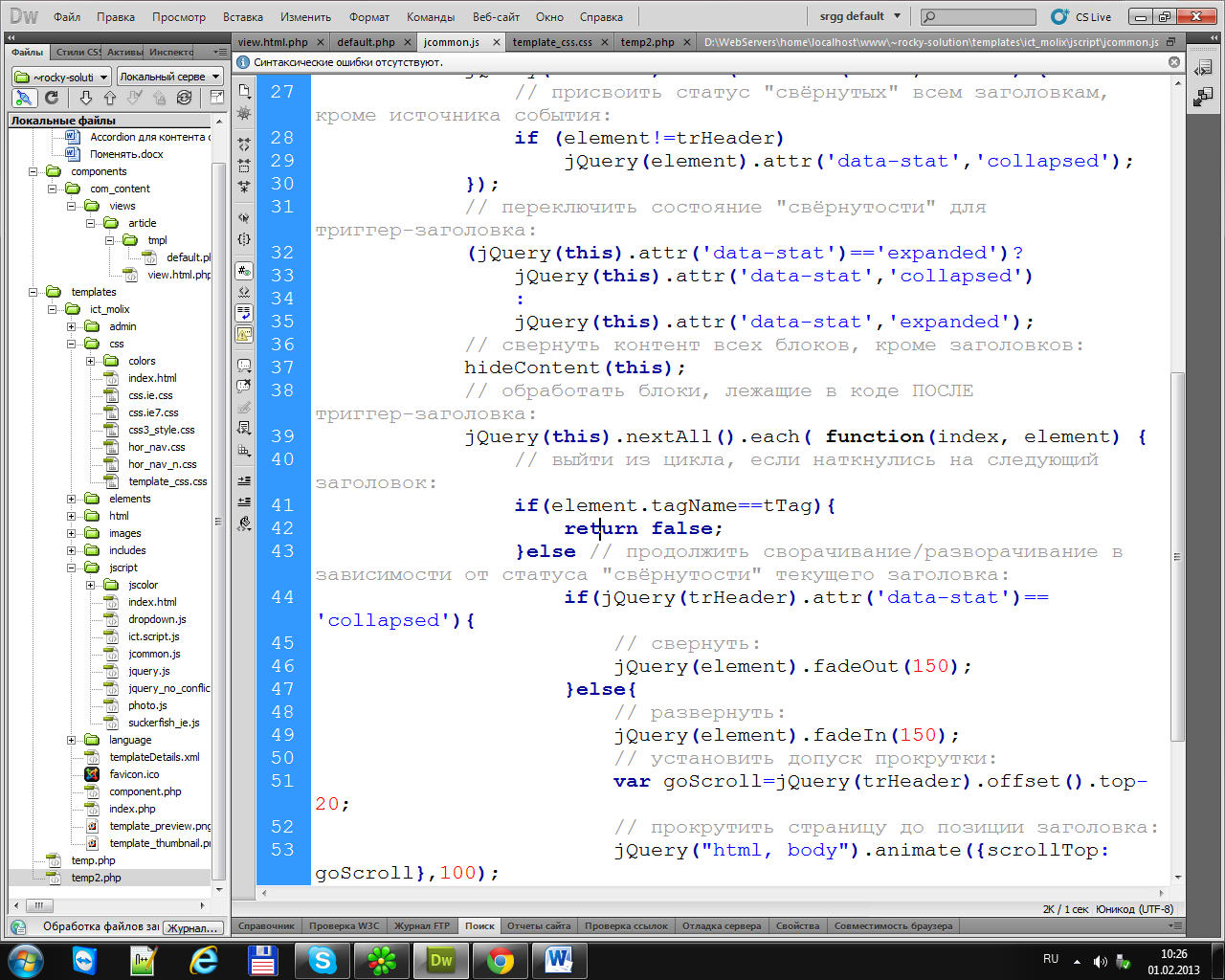
1. Сворачивает все текстовые блоки. Для этого вызывается функция hideContent().

#### При возникновении события (клик по заголовку)

1. Определяет источник события (trigger-объект).
2. Всем заголовкам, кроме источника события, назначается статус «свёрнутых» (data-stat=”collapsed”), что определяет соответствующее состояние текстовых блоков. Данный параметр также определяет вид заголовков (с ним ассоциированы соответствующее CSS-правила в основной таблице стилей шаблона).
3. Последовательно проходит по всем DOM-элементам, находящимся не выше в его иерархии, чем trigger-объект и находящихся после него в коде, и разворачивает их до тех пор, пока не встретит следующий DOM-объект с [тем же тегом](#default_tag), или с тегом, [добавленным в исключения](#exception_tag).
4. Также процедура обработки теста работает как переключатель — если элементы целевого объекта были свёрнуты — разворачивает их, и наоборот. Контроль состояния блоков осуществляется опосредованно, в соответствии с атрибутом «data-stat» заголовка (**expanded** или **collapsed**):



…



### Настройка jQuery-обработчика и его подключение

Вы ничем не ограничены в настройке данного скрипта за исключением описанной выше базовой логики его работы. Обработчик находится по адресу:

/templates/ict\_molix/jscript/jcommon.js

…и подключён в файле шаблона.

**Внимание!** При изменении настроек скрипта требуется использовать имя *jQuery* вместо сокращённого варианта *$*, поскольку в последнем случае возникает ошибка (вероятно, в следствие несовместимости скриптов, относящихся к обработке шаблона).

Или использовать $ в качестве ссылки на объект jQuery:

jQuery(function($){ // code });

## Напутствие

Удачи. И да поможет вам ВР!