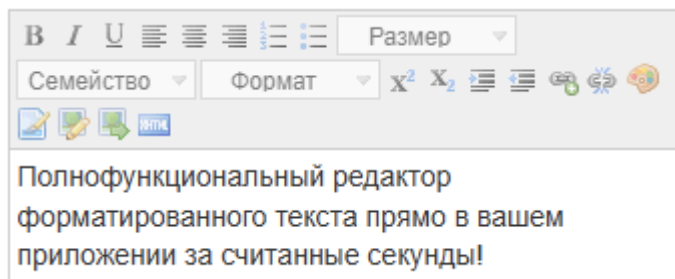


NicEdit

NicEdit — это облегченный, кроссплатформенный редактор встроенного контента, позволяющий легко редактировать содержимое веб-сайта в режиме реального времени прямо в браузере.



NicEdit Javascript интегрируется в любой сайт за считанные секунды, позволяя сделать любой элемент/div редактируемым или преобразовать стандартные текстовые поля в редактор форматированного текста.

Документация

This is the official manual for the NicEdit WYSIWYG Editor. NicEdit is a Lightweight, Cross Platform, Inline Content Editor to allow easy editing of web site content on the fly in the browser. To download it go to <http://nicedit.com/>

- [About NicEdit](#)
- **NicEdit Reference**
 - [Javascript API](#)
 - [Configuration Options](#)
- **Plugin Reference**
 - [Saving via AJAX](#)
 - [XHTML Compliant Output](#)
- **Customization**
 - [Developing with NicEdit](#)
 - [Creating a Plugin](#)
 - [Integration with NicEdit events](#)
 - [How to translate nicEdit \(with Spanish example\)](#)

About NicEdit

NicEdit — это лёгкий, кроссплатформенный редактор встроенного контента, позволяющий легко редактировать содержимое веб-сайта на лету прямо в браузере. Этот небольшой и простой в использовании JavaScript-редактор интегрируется в любой сайт за считанные секунды, делая любой элемент/div редактируемым или преобразуя стандартные текстовые поля в форматированный текст.

NicEdit был разработан Брайаном Кирхоффом и может свободно использоваться в любых целях под лицензией MIT.

Функции:

- Небольшой размер файла (<35 КБ в сумме), <10 КБ в сжатом виде!

- Для работы требуется всего 2 файла (js + иконки).
- Гибкая конфигурация: замена текстовых полей или блоков div.
- При желании несколько редакторов могут использовать один элемент управления.
- Сохранение контента с помощью AJAX или HTTP POST
- IE 5.5+ / FF 2+ / Opera 9+ / Safari 3+

NicEdit Reference

Javascript API

Class nicEditors

The nicEditors class provides convenience methods to add NicEditor to the page or get a reference to a editor by ID

<code>nicEditors.allTextAreas</code>	Converts all the textareas on the page into NicEditor instances. Returns a reference to the array of all NicEdit instances on the page.
<code>nicEditors.findEditor</code>	Finds a specific editor by ID if created using allTextAreas explicitly. For example to get the nicedit instance for <code><textarea id="myArea2"></textarea></code> use: <code>nicEditors.findEditor('myArea2');</code> The method returns a nicedit instance that you call the functions below on. Warning: In order to use this and other API methods, the editor must be finished loading. To see an example of how you can use addEvent to have your code run after editor load see Editor Events
<code>nicEditors.editors[]</code>	The raw array of all the NicEdit instances on the page

Class nicInstance

The nicEditor instance class creates the editable area for a single element and provides a number of useful methods to developers. You can get a reference to a specific nicInstance object using `nicEditors.findEditor`

<code>[nicInstance].getContent()</code>	Returns the current HTML of the nicInstance. For example: <code>nicEditors.findEditor('myArea2').getContent();</code> returns the HTML in the content editor that replaced the element on the page with ID 'myArea2'.
<code>[nicInstance].setContent(HTML)</code>	Set the current HTML in the editor instance. For example: <code>nicEditors.findEditor('myArea2').setContent('Some HTML here');</code>
<code>[nicInstance].saveContent()</code>	Only for nicInstances that are replacing a <code><textarea></code> this method syncs the content of the editor with the textarea value. This is done automatically if the form with the original <code><textarea></code> is submitted. However, you may want to explicitly do the syncing yourself.

Class nicEditor

The nicEditor class is a container for a number of nicEditor instances and 1 nicPanel. Because of this

you can use nicEditor in 2 ways:

1. Create a single control panel with `setPanel()` then use `addInstance` multiple times to create as many editable areas on the page you need that are controlled from a single control panel.
2. Use the `panelInstance()` to create a combined content area/panel editor. If you use this option and want multiple editors you should create multiple nicEditor instances.

<code>var myNicEditor = new nicEditor()</code>	Creates a new nicedit object. A single instance of nicEditor contains:- 1 or more editor instances (nicInstance/nicFrameInstance) - 1 nicPanel
<code>myNicEditor.addInstance('someElementsID')</code>	Adds a nicedit instance to allow inline editing of the given elements ID
<code>myNicEditor.removeInstance('someElementID')</code>	Removes the nicedit instance with the given ID, returning it to the original element with the content within the element.
<code>myNicEditor.setPanel('someElementID')</code>	Append an editor control panel at the end of the given ID. Normally the element is blank and with a fixed CSS width you would like the panel set.
<code>myNicEditor.panelInstance('someElementID')</code>	Creates an inline content editor with attached control panel on top of the element.
<code>myNicEditor.instanceById('someElementID')</code>	Find a nicInstance object by ID that was added to this nicEditor
<code>myNicEditor.floatingPanel()</code>	Activate a floating panel. First add editor instances to this editor with <code>addInstance()</code> then call <code>floatingPanel()</code> . When a instance is selected, a floating panel will appear above/below it. * Requires nicFloating plugin

Configuration Options

NicEdit обладает широкими возможностями настройки: параметры конфигурации передаются при создании объекта NicEditor. Передавайте конфигурацию при вызове: `new nicEditor({CONFIG HERE})`

Добавьте `.panelInstance('ID TO TEXTAREA HERE')`, чтобы добавить редактор в текстовое поле.

Вы можете указать несколько параметров конфигурации, они представлены в формате JSON: `{fullPanel : true, iconsPath : 'nicEditorIcons.gif', maxHeight : 200}`

Обратите внимание, что некоторые параметры конфигурации, такие как список кнопок, не будут работать, если вы не загрузите версию NicEdit и не используете её на своём веб-сервере.

Доступные параметры конфигурации:

fullPanel	Если установлено значение true, будет создан экземпляр редактора со всеми доступными кнопками. Если значение false (по умолчанию), для управления отображением кнопок будет использоваться buttonList.
buttonList	Например, массив кнопок в редакторе. ['жирный','курсив','подчеркнутый','слева','по центру']
iconsPath	Строковый путь к файлу значков, например, images/nicEditorIcons.gif.
maxHeight	Высота в пикселях, за пределы которой редактор не должен автоматически расширяться. Область редактирования может быть больше указанного значения, но в этом случае будут использоваться полосы прокрутки. Для редактора с фиксированной высотой (не динамически увеличивающегося) установите значение maxHeight и style=«height: X» равным одному и тому же значению в пикселях.
externalCSS	Относительный путь к внешней таблице стилей, применяемой к экземплярам iframe nicEditor (этот режим работы используется только в FF2). Пример: {externalCSS : 'mysite.css'}
uploadURI	Абсолютный или относительный URL-адрес скрипта-приемника nicUpload. Позволяет загружать изображения из nicedit на ваш собственный сервер. *Требуется nicUpload Пример:{uploadURI : 'http://yourdomain.com/nicUpload.php'}. По умолчанию (http://files.nicedit.com/) используется сервер Nicedit, а файлы загружаются на Imageshack. PHP-версию скрипта для загрузки файлов можно найти по адресу: http://nicedit.com/svn/nicedit/trunk/nicUpload/php/nicUpload.php

List of Buttons for buttonList option

- 'bold'
- 'italic'
- 'underline'
- 'left'
- 'center'
- 'right'
- 'justify'
- 'ol'
- 'ul'
- 'subscript'
- 'superscript'
- 'strikethrough'
- 'removeformat'
- 'indent'
- 'outdent'
- 'hr'
- 'image'
- 'upload' * requires nicUpload
- 'forecolor'
- 'bgcolor'
- 'link' * requires nicLink
- 'unlink' * requires nicLink
- 'fontSize' * requires nicSelect
- 'fontFamily' * requires nicSelect
- 'fontFormat' * requires nicSelect

- 'xhtml' * required nicCode

Plugin Reference

Saving via AJAX

NicEdit поддерживает наличие кнопки сохранения на панели инструментов, которую пользователи могут нажать для сохранения контента.

Для использования этого режима вам потребуется базовое знание JavaScript, и вы должны выполнить следующие шаги:

- Сначала убедитесь, что вы настроили загрузку с помощью плагина nicSave.
- Используйте параметр onSave при объявлении экземпляра nicEditor, например:

script.js

```
bkLib.onDomLoaded ( function ( ) {  
    new nicEditor ( { fullPanel: true , onSave: function ( content, id,  
instance ) {  
        alert ( 'Кнопка сохранения нажата для элемента ' + id + ' = ' + content  
    ) ;  
    } } ) . panelInstance ( 'myArea2' ) ;  
} ) ;
```

- Поскольку мы установили параметр onSave для нашей функции обратного вызова и добавили кнопку «Сохранить», значок сохранения появится на панели инструментов.
- При нажатии кнопки будет запущена функция обратного вызова onSave с тремя параметрами.
 - Первый параметр — это содержимое экземпляра nicEditor.
 - Второй параметр — это идентификатор элемента, преобразованного в nicEditor.
 - Третий параметр — это ссылка на nicInstance (подробнее об этом см. в документации по [Javascript API](#)).

XHTML Compliant Output

Customization

Developing with NicEdit

Creating a Plugin

Integration with NicEdit events

How to translate nicEdit (with Spanish example)

Дополнения и Файлы

- Сжатый NicEdit (без пробелов и комментариев). Рекомендуется для развертывания NicEdit.
- Несжатая версия NicEdit для разработки. Используйте её только для отладки или разработки NicEdit.

From:
<https://wwoss.direct.quickconnect.to/> - worldwide open-source software

Permanent link:
<https://wwoss.direct.quickconnect.to/doku.php?id=software:development:web:docs:web:wysiwyg:nicedit&rev=1767787150>

Last update: **2026/01/07 14:59**

