

# CODE PER NODE MODULE: per-node and per-block enhancements via custom CSS and Javascript

## I. SETUP

Drupal module download link: [Code Per Node](#) (CPN)

NOTE: The following presupposes a CKEditor environment via the [CKEditor module](#) and version 4.1.x of CKEditor. There is another way to configure the CKEditor via the WYSIWYG module, but this did not expose the Advanced settings that allows one to disable filtering of HTML tags.

### In the Drupal UI, setup of the module for optimal use

0) Install and enable CPN module, it will appear in the Other section of the Modules page.

ENABLED	NAME	VERSION	DESCRIPTION	OPERATIONS
<input checked="" type="checkbox"/>	Advanced help	7.x-1.0	Allow advanced help and documentation. Required by: Chaos Tools (CTools) Plugin Example (disabled), Advanced help example (disabled)	<a href="#">Help</a> <a href="#">Permissions</a>
<input type="checkbox"/>	Advanced help example	7.x-1.0	A example help module to demonstrate the advanced help module. Requires: Advanced help (enabled)	
<input checked="" type="checkbox"/>	Backup and Migrate	7.x-2.7	Backup or migrate the Drupal Database quickly and without unnecessary data.	<a href="#">Help</a> <a href="#">Permissions</a> <a href="#">Configure</a>
<input checked="" type="checkbox"/>	Code per Node	7.x-1.4	Manage custom CSS & Javascript per node and per block.	<a href="#">Permissions</a> <a href="#">Configure</a> <span style="border: 1px solid red; border-radius: 50%; padding: 2px;">1</span>

1) Via Configure 1, set the directory for CSS and JS files created by the CPN module, also enable/disable Syntax Highlighting (requires manual download of CodeMirror (<http://codemirror.net/codemirror.zip>) and placement into the location stated in b) below.

<http://example.com/admin/config/content/cpn>

Home » Administration » Configuration » Content authoring

### Code per Node o

**Syntax Highlighting**

None

CodeMirror 2

**File storage path**

The path where Code per Node will store files. Note: changing this path will cause the old path to move to the new path, overwriting the new path, if it exists.

## 2) Set desired Permissions for CPN

<http://example.com/admin/people/permissions#module-cpn>

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
<b>Code per Node</b>			
Edit CSS per node	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Edit JavaScript per node	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Edit CSS per block	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Edit JavaScript per block	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## 3) Adjustment for CKEditor to make HTML content not be filtered, a customization of the CKEditor configuration at:

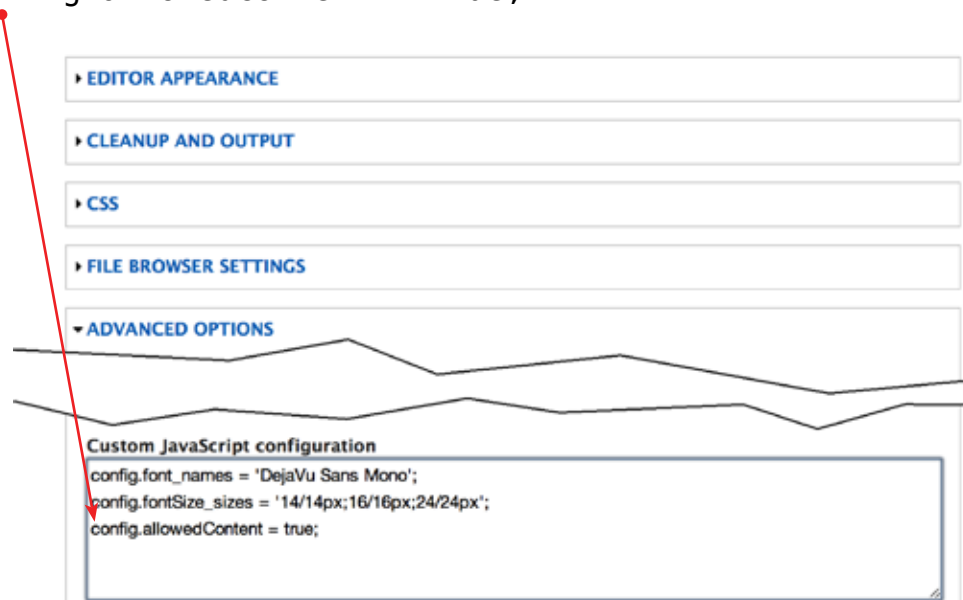
<http://example.com/admin/config/content/ckeditor>

Then edit either the Full or Filtered HTML, e.g. Full:

<http://example.com/admin/config/content/ckeditor/edit/Full>

In the Advanced Options in the Custom Javascript configuration field, add the following line:

```
config.allowedContent = true;
```



Some additional notes on this are mentioned on the module page for the [CKEditor module](#):

*Note for CKEditor 4.1+ users*

*If you are using CKEditor 4.1+, you should adjust Advanced Content Filter settings. If you have any trouble implementing this feature, you can turn it off by setting:*

```
config.allowedContent = true;
```

in the Custom JavaScript configuration field of the Advanced Options section for your CKEditor profiles.

4) CPN settings for Content Types , enable (or disable) the option to add CSS and Javascript to nodes

<http://example.com/admin/structure/types>

e.g. Basic Page - <http://example.com/admin/structure/types/manage/page>

The screenshot shows the Drupal administration interface for editing the 'Basic page' content type. The breadcrumb trail is 'Home » Administration » Structure » Content types'. The page title is 'Basic page' with a dropdown arrow. There are five tabs: 'EDIT' (active), 'MANAGE FIELDS', 'MANAGE DISPLAY', 'COMMENT FIELDS', and 'COMMENT DISPLAY'. Below the tabs, the 'Name' field is set to 'Basic page' with a machine name of 'page'. A description field contains the text: 'Use <em>basic pages</em> for your static content, such as an 'About us' page.' Below this is a larger text area for describing the content type. On the left, a sidebar contains several settings sections: 'Submission form settings' (Title), 'Publishing options' (Published), 'Display settings' (Don't display post information), 'Comment settings' (Hidden, Threading, 50 comments per page), 'Code per Node settings' (highlighted), 'Menu settings', and 'Read More Settings'. The main content area shows 'CSS' and 'Javascript' sections, each with a text area for custom rules. Below these are two checked checkboxes: 'Enable custom CSS per node' and 'Enable custom Javascript per node', each with a description of the permission required for users to edit these rules.

## Locations on the file system (server or local machine) of note

a) e.g., my directory for CPN .css and .js files, this is user-specified in the UI in item 1)

```
/sites/all/css
```

NOTE: .css and .js files are named with the number of the node they are associated with, e.g. 27.css is the css for the page at <http://example.com/node/27>

b) Where you place CodeMirror for syntax highlighting:

```
/sites/all/libraries/codemirror
```

## Helpful material for CSS and Javascript work

JSFiddle: an online CSS and Javascript experimentation tool:

<http://jsfiddle.net>

Interesting examples of CSS and Javascript to experiment with:

<http://andrewwooldridge.com/blog/2011/03/16/stunning-examples-of-using-jsfiddle/>

Tips on working with Javascript and jQuery in Drupal:

<https://drupal.org/node/171213>

In the above link, the notes (heading entitled **Javascript closures**) on the concept of closure in Javascript and jQuery is worth reading as this will help with not getting undefined function alerts. In general, if you are working with functions (beyond just calling them), wrapping your code in the following should help, e.g.:

```
(function ($) {  
    your code here  
})(jQuery);
```

## II. EXAMPLES

Drop shadow (CSS `text-shadow`) applied to `p` and `h1` tags, as seen in the editor for a basic page, results on next page.

The screenshot displays a web editor interface. At the top, there's a toolbar with various icons for editing, including undo, redo, search, and text alignment. Below the toolbar is a rich text editor with buttons for bold (B), italic (I), underline (U), strikethrough (ABC), and text color (A). The main editing area contains two paragraphs of placeholder text. Below the editor is a 'Text format' section with a dropdown menu set to 'Filtered HTML' and a list of allowed HTML tags. The 'Image Insert' section includes a 'Choose File' button and an 'Upload' button. On the right side, there's a 'Menu settings' sidebar with options like 'Code Per Node example', 'CSS & Javascript', 'Revision information', 'URL path settings', 'Comment settings', and 'Authoring information'. The 'CSS & Javascript' section is active, showing a code editor with the following CSS rules:

```
h1
{
color:#dfd;
text-align:center;
text-shadow: 2px 2px #335533;
}
p
{
color:#000;
font-family:'Optima';
font-size:20px;
text-shadow: 1px 1px #eee;
```

How drop shadow appears:

The screenshot shows a web application interface for 'bluelichen.dev'. At the top left is a logo featuring green lichen-like structures. The main header contains the site name 'bluelichen.dev' and a navigation menu with items: Home, Code Per Node example (active), R Shiny example, jsfiddle test - JQuery Heat!, Blue Heat, and test classes. Below the navigation is a search bar with a magnifying glass icon and a breadcrumb trail: Home > Code Per Node example. On the left side, there is a 'Navigation' section with links for 'Add content' and 'Recent content'. The main content area features a large heading 'Code Per Node example' in green, followed by three buttons: 'View' (active), 'Edit', and 'Devel'. The content consists of three paragraphs of placeholder text in Latin.

# bluelichen.dev

Home Code Per Node example R Shiny example jsfiddle test - JQuery Heat! Blue Heat test classes

Home > Code Per Node example

## Code Per Node example

View Edit Devel

Nulla vitae elit libero, a pharetra augue. Cras mattis consectetur purus sit amet fermentum. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Vestibulum id ligula porta felis euismod semper. Nullam id dolor id nibh ultricies vehicula ut id elit. Nullam quis risus eget urna mollis ornare vel eu leo. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Maecenas faucibus mollis interdum. Praesent commodo cursus magna, vel scelerisque nisl consectetur et. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum. Nulla vitae elit libero, a pharetra augue.

Donec ullamcorper nulla non metus auctor fringilla. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Donec ullamcorper nulla non metus auctor fringilla. Nullam id dolor id nibh ultricies vehicula ut id elit. Etiam porta sem malesuada magna mollis euismod.

Simulation of fire and smoke on the `p.fire` tag, then ice on the `p.ice` tag, the code as seen in the editor and the results on the next page.

The screenshot shows a web editor interface. At the top, there's a toolbar with various icons for editing. Below the toolbar is a text area containing the following text: "Text at -900°C", "H<sub>2</sub>O at 0°C", and a small graphic of a red flame with three black smoke plumes. Below the text area is a "Switch to plain text editor" link. Underneath that is a "Text format" dropdown menu set to "Full HTML" and a link for "More information about text formats". Below the text format section is an "Image Insert" section with a "Choose File" button (showing "no file selected") and an "Upload" button. Below the image insert section is a note: "Files must be less than 32 MB. Allowed file types: png gif jpeg jpeg". At the bottom of the editor is a sidebar with several sections: "Menu settings" (test classes), "CSS & Javascript" (highlighted), "Revision information" (No revision), "URL path settings" (Automatic alias), "Comment settings" (Closed), "Authoring information" (By robw on 2013-07-12 12:57:28 -0700), and "Publishing options" (Published). The "CSS & Javascript" section is expanded, showing a CSS rule for `p.ice` with the following properties: `text-align:center;`, `font:900 120px/150px Arial;`, `/* padding-top:150px; */`, `color:#9bbbed4;`, `white-space:nowrap;`, and `text-shadow:` followed by a list of shadows: `-2px -1px 2px DarkCyan,`, `2px -2px 2px SteelBlue,`, `2px -10px 20px white,`, `0 -2px 6px PowderBlue,`, `2px -2px 10px white,`, and `5px -10px 15px LightCyan,`. Below the CSS code is a note: "Custom CSS rules for this node. Do not include <style> tags. Enable syntax highlighting."

Complete CSS' listed on the following page for experimenting with. The fire CSS is from sidonaldson (flaming text) on this site:

<http://andrewwooldridge.com/blog/2011/03/16/stunning-examples-of-using-jsfiddle/>

```

p.ice{
  text-align:center;
  font:900 120px/150px Arial;
  /* padding-top:150px; */
  color:#9bbbed4;
  white-space:nowrap;
  text-shadow:
    -2px -1px 2px DarkCyan,
    2px -2px 2px SteelBlue,
    2px -10px 20px white,
    0 -2px 6px PowderBlue,
    2px -2px 10px white,
    5px -10px 15px LightCyan,
    8px -20px 20px Cyan,
    12px -30px 25px DodgerBlue,
    5px -60px 40px DodgerBlue,
    15px -34px 30px DeepSkyBlue,
    0px -36px 35px Aquamarine,
    40px -40px 50px Aquamarine,
    -20px -60px 40px Aquamarine,
    20px -80px 40px Aquamarine,
    40px -80px 40px Aquamarine,
    20px -120px 60px Aquamarine,
    20px -150px 80px LightSkyBlue,
    50px -160px 80px LightSkyBlue,
    100px -200px 100px LightSkyBlue,
    120px -150px 100px LightSkyBlue,
    100px -100px 100px LightSkyBlue,
    120px -220px 100px LightSkyBlue,
    200px -300px 150px LightSkyBlue
  ;
}

```

```

p.fire{
  text-align:center;
  font:900 120px/150px Arial;
  padding-top:150px;
  color:#000;
  white-space:nowrap;
  text-shadow:
    -2px -1px 2px purple,
    2px -2px 2px blue,
    2px -10px 20px white,
    0 -2px 6px red,
    2px -2px 10px white,
    5px -10px 15px lightyellow,
    8px -20px 20px yellow,
    12px -30px 25px orange,
    5px -60px 40px orange,
    15px -34px 30px darkorange,
    0px -36px 35px red,
    40px -40px 50px red,
    -20px -60px 40px red,
    20px -80px 40px red,
    40px -80px 40px red,
    20px -120px 60px red,
    20px -150px 80px black,
    50px -160px 80px black,
    100px -200px 100px black,
    120px -150px 100px black,
    100px -100px 100px black,
    120px -220px 100px black,
    200px -300px 150px black
  ;
}

```



The visual results within Drupal 7:

