Jtricks.com

Fixed'n'Sticky

Module for Joomla!

Version 1.2.1

Preface

The module being placed into one of existing module positions creates a new position for module(s) (called target modules in this document) within a DIV block container that will move with the user scrolling (using **position:fixed** as much as possible) using javascript.

Compared with straight position:fixed usage, this module will confine the container to the boundaries of module position, such as **left** or **sidebar-a** in some templates.

The way module integrates into Joomla templating system can be represented in the following way:
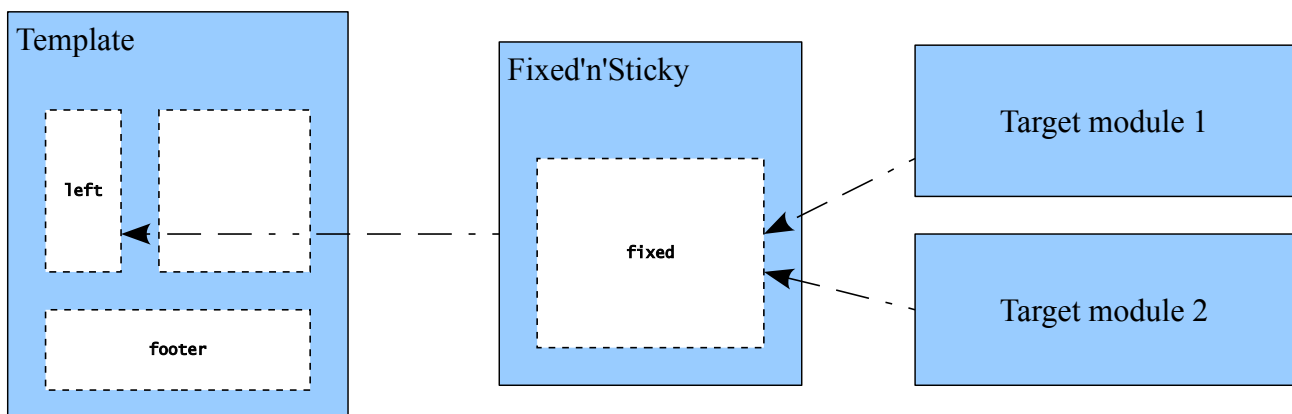
Several copies of the module can be used to to provide several module locations within respective fixed DIV boxes.

Joomla versions supported:

> Joomla! 1.5 (tested on 1.5.22);
> Joomla! 1.6 (tested on 1.6.0);
> Joomla! 1.7 (tested on 1.7.2);
> Joomla! 2.5 (tested on 2.5.6);
> Joomla! 3.0 (tested on 3.0.0).

There is a special support for module chrome handling for:

- Artisteer templates;

- Joomlashine (joomlashine.com) templates;

- Support for WARP-based templates (e.g. YOOtheme); read separate chapter in this documentation; Tested against versions 5.5.14 and 6.0.8 of the engine.

How to use it the module

To use the Fixed'n'Sticky module:

- Install the module using the extension manager (archive root directory contains module version for Joomla! 1.6 and above, j15 directory contains module version for Joomla! 1.5);

- Configure your **Fixed'n'Sticky** module (please note that it is just a container for other modules):

- Put Fixed'n'Sticky in module position **existing in template**.

- **Enable it** (selecte Enabled – Yes on Joomla 1.5 or Published –

published on Joomla 1.6);

- **Perform menu assignment** to the menu items you wanted your target module to be shown;

- Normally you don't want to show its title (Show Title: Hide);

- In certain cases you might need to override CSS style if your menu shows incorrectly. In this case you can use '#fixeddiv_fixed' CSS selector as this will be the id of the container (unless 'Module position to provide' is changed in Fixed'n'Sticky parameters).

- Configure your **desired target (child) module** (such as main menu module) :
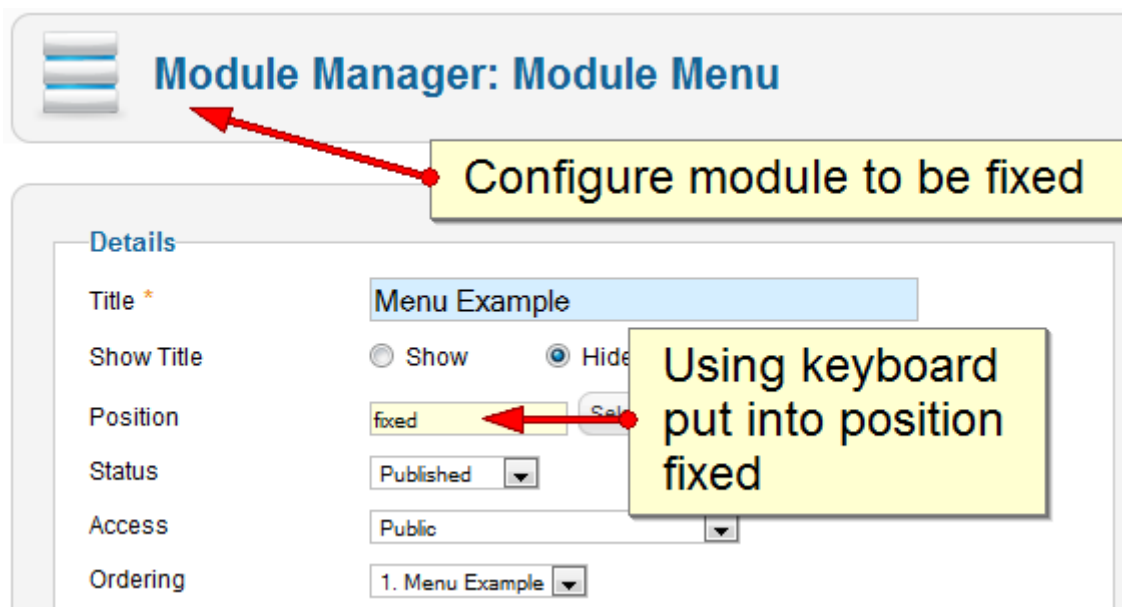
- Put your desired target module to be fixed to position '**fixed**' (or what 'Module position to provide' parameter in Fixed'n'Sticky parameters is changed to) - this needs to be entered manually into the position as it is not provided by template and hence isn't present in its config;

Parameters description

Default prameter values are configured to be of use in most often used configurations.

Parameter	Description
Basic parameters	
Debug	If enabled, will show a grey box with some debug information that will help solve integration problems.
Module position to provide	The module will provide new position and this is the name of the position. It needs to be changed if several copies of the module are used to have multiple fixed boxes on screen – it must be unique.
Distance from the top of the window	Distance from the top or bottom of the window. Only one of these should be specified.
Distance from the left border of the window	In the same way as Y-axis distances, represent distance from the left or right border of the window. Only one of these should be specified.
Auto-adjust column height for short pages in confined mode	On some templates HTML code produced by module might display overlapping content below when component (article) is too short.
If height auto-adjust stops floating, confinement area class or confinement area id	In confined mode the module will try to guess the area it should stay in by navigating DOM hierarchy upwards looking for container with the size to fit the fixed box. In some situations (and especially if auto-adjust column height is enabled) it will be unable to find correct confinement area. To solve this, the confinement area HTML tag can be manually specified by indicating its id or class.
Advanced parameters	
CSS style overrides (will be placed into html HEAD)	Can include CSS definitions that will be placed into page header only on the pages the module is present. Will allow to make template CSS overrides. Fixed container style can be adjusted by using **#fixeddiv_fixed** (or rather **#fixeddiv_<name of the module position>**, if the position name was changed) selector.
Module inclusion style (template-dependant)	Normally, Fixed'n'Sticky will use the module

	inclusion style (or module chrome) from the position it is in for the target modules. This parameter when not empty will override that. Module inclusion style is a very template-dependant parameter. Most often used values are: - xhtml - none - rounded To see which styles are used to display contents by your template append **?tp=1** to your URL. This will show the page overlayed by module location names and inclusion style. For example, **rhuk_milkyway** template uses **rounded** module inclusion style to display rounded corners in modules such as mainmenu in **left** module position.
HTML prelude for modules HTML finale for modules	Prelude and finale to be used whene rendering modules. It can be used for example when there should be a border around all the child modules, not around each child module separately. For example: prelude finale Will paint red border around all child modules as a whole.

Using Fixed'n'Sticky module with WARP-based templates (from YOOtheme)

YOOtheme templates use special templating engine (WARP) which deviates significantly from how Joomla usually renders modules.

In order to use Fixed'n'Sticky with warp-based templates, Fixed'n'Sticky and target module(s) to be fixed must be configured in a special way.

Target module(s) to be fixed:

- Set Module Class Suffix advanced parameter to 'style-<module style>', where <module style> is a style suggested by your template documentation or the one selected for your module position in file <template dir>/layouts/module.php. For example, 'style-line' or 'style-box' etc.

Fixed'n'Sticky:

- Set Module Class Suffix advanced parameter to 'style-raw'