

Log in

Find

Pre-Tiki 13 CSS transition to the Bootstrap Framework Stylesheets

This page is about Tiki's transition to using the Bootstrap CSS Framework, and specifically about the organization and use of CSS files.

Pre-Tiki 13 tasks

- Revamp of pre-Tiki 13 stylesheet organization
 - Switch to using bootstrap.css rules as much as possible, to reduce the number of rules in design.css and the css directory files
 - Rename design.css to tiki.css
Suggested by LP, this is because the purpose of the file is not to provide design rules, but to enable legacy functions and other functions outside the scope of bootstrap.css, such as show/hide display rules.
 - Rules in design.css currently:
 - Position modules - admin module, etc.
 - Position elements in middle center of page - wiki action icons, etc.
 - Provide wikiplugin and module style details.
 - Provide feature layout and styling
 - etc. (identification in progress)
 - Replace color rules in legacy CSS files with Bootstrap colors as much as possible
This is to reduce rule redundancy and to avoid color contrast conflicts and improve color consistency (aesthetic and information-wise).
- Update packaged theme files (those that would be an asset)
It's fairly straightforward to make the files in styles/ compatible with Bootstrap, so these will mostly be kept and updated (see below).
- Maybe only the themes in mods.tiki.org that people request to be updated will in fact be updated; IOW,

only do work where there is demand.

- Document how to convert a (WP, Joomla or other) Bootstrap theme to use with Tiki
Some conversion projects prior to Tiki 13 release will give some information on this process.

Theme administration (Look & Feel) mini-revamp

A smooth process is needed, to support various theme situations:

1. Tiki.css (design.css renamed) should be loaded in all cases.
2. Using an included Tiki theme - no change from previous Tiki versions - these will be bootstrap.css variants: choose with dropdown selector. (Any external file can also be put in the 'styles' directory and chosen this way.)
3. Using an external theme bootstrap.css file
 - Users should have the option to use a custom bootstrap.css file instead of the default one: text input to indicate file location. They could ftp it into the styles directory, or input the path.
 - This theme file upload and file switching needs to be in L & F admin



Screenshot with this added pref:

- Using an external file that supplements the bootstrap.css file (in the same way that a pre-Tiki13 theme file supplements design.css) - process similar to that in previous Tiki versions: In this method, the default bootstrap.css needs to be loaded first, then the theme stylesheet. Of course, here too tiki.css is also loaded.

4. About changing the layout template to match the style (wider side columns, one or two side columns, no side columns, etc.), this was formerly possible with theme stylesheet and sometimes theme-specific tiki.tpl; now a switch needs to be made in templates/layouts/ because the grid column width classes are hard-coded in layout_view.tpl.

Making a Tiki theme as a bootstrap.css variant

Theme files were committed as part of working out a method for making Tiki themes using the Less CSS pre-processor and a copy ("custom-variables.less") of the Bootstrap variables.less file. This is about the themes in the Tiki package, but the same process can be used to make Tiki themes independently. The variables are edited as needed to get the look and feel for the theme. For elements and properties, etc. that are outside the scope of bootstrap.css, there's a "custom-other.less" file. These are in the "styles/newtheme" directory. The "styles/newtheme.less" file imports the variables files and it's all compiled to produce "newtheme.css".

This method is described by Thomas Park, who does the Bootswatch themes, at <http://coding.smashingmagazine.com/2013/03/12/customizing-bootstrap/>.

The "newtheme.css" and other CSS files shouldn't be edited directly. All editing for the theme is done in the "styles/newtheme/*.less" files. PhpStorm handles the compiling transparently. The CSS files are produced or

updated automatically after each .less file change. Or another method can be used to watch for edits and compile. Check Less pre-processor references online.

So far (April 20, 2014), the Darkroom, Ohia, and TheNews themes have been converted in this way to bootstrap.css variants.

About using design.css, the Look and Feel admin process is still being worked out to make selecting the theme a smooth process.

Regarding the Tiki project files workflow and Less, I suggest that both the Less files and the compiled CSS be committed. See <http://jeffcroft.com/blog/2012/feb/23/many-ways-to-use-css-preprocessors/> for more on this.

Tasks related to CSS and themes

- Tiki file system revamp (this isn't necessary for Tiki 13 so will probably be delayed):
 - Consolidate location of CSS files
 - Consolidate theme files (.css and .tpl, etc.) in one directory
 - Have all "custom" files in one place
- Docs for optimizing responsiveness of your site
 - Using grid divs in wiki text areas
 - Handling wide images and tables, etc.
 - Using modules, etc. to achieve Bootstrap layouts

Alias

[BS css](#)