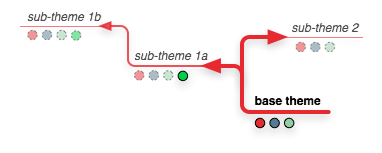
Creating a sub-theme

*Last updated May 29, 2015. Created on February 22, 2008.  
Edited by [Willynux](https://www.drupal.org/u/willynux" \o "View user profile.), [JakeRogers](https://www.drupal.org/u/jakerogers" \o "View user profile.), [NarendraR](https://www.drupal.org/u/narendrar" \o "View user profile.), [KCarlile](https://www.drupal.org/u/kcarlile" \o "View user profile.).*[*Log in to edit this page*](https://www.drupal.org/user?destination=node/225125)*.*

Sub-themes are just like any other theme, with one difference: They inherit the parent theme's resources. There are no limits on the chaining capabilities connecting sub-themes to their parents. A sub-theme can be a child of another sub-theme, and it can be branched and organized however you see fit. This is what gives sub-themes great potential.



Imagine starting with a base theme designed as wireframes, then applying and refining all the details from a sub-theme. Then, from the same wireframe, testing out alternate designs by branching out another sub-theme. Working on a multi-site installation but you need a cohesive look and feel? With sub-theming, a lot of the design resources can be shared. Site-specific changes can be set to a specific sub-theme, but any shared resources can be edited once to be applied across all the installations. With careful planning, the possibilities are endless.

**Creating a sub-theme**

A sub-theme must have a different internal name (i.e., the name used by Drupal) from its parent theme. This name must not contain any spaces or other special characters. **The name of your sub-theme must start with an alphabetic character and can only contain lowercase letters, numbers and underscores**. Let's suppose this sub-theme internal name is "my\_subtheme".

**Folder *my\_subtheme***

The sub-theme to-be should be located in its own directory. This folder should have the same name as the internal name of the sub-theme (e.g., *my\_subtheme*).

The sub-theme folder should be located inside the folder sites/*example.com*/themes/ ("*example.com*" stands for your site name) or, if you want to use your sub-theme for several sites, inside sites/all/themes/ (e.g., sites/*example.com*/themes/*my\_subtheme* or sites/all/themes/*my\_subtheme*).

**File *my\_subtheme*.info**

To declare your theme to be a sub-theme of another, you must put a *my\_subtheme*.info file inside your my\_subtheme folder (remember that "*my\_subtheme*" stands for your sub-theme internal name). The easiest way is usually to copy the *theme\_name*.info file from the parent theme ("*theme\_name*" stands for the internal name of the parent theme), and to rename it as *my\_subtheme*.info. Then you need to add the following line to*my\_subtheme*.info file to declare its parent or "base theme.": change "theme\_name" to the internal name of the parent theme (that is, the name of the parent theme's .info file, usually all lower case).

base theme = theme\_name

It is also a good idea to modify the human-readable name of your sub-theme by altering the line beginning by name = , and the description of your sub-theme by altering the line beginning by description =  . For instance:

name = my\_subtheme  
description = This is a sub-theme of theme Bartik, made by John for the web site example.com (red, responsive).  
core = 7.x  
base theme = bartik

As the sections below indicate, the sub-theme inherits *most* properties of the base theme. The important exceptions are regions, core version, and color info. You probably want to copy the regions section of your base theme's info file, along with its core version declaration. If your base theme supports the color module and you'd like your sub-theme to support it, you probably also want to copy the color folder from your base theme and add the line from your base theme's info file to your sub-themes info file that looks like

stylesheets[all][] = css/colors.css

and then copy the colors.css from your base theme to the css folder of your sub-theme.

**Style sheet inheritance**

All [style sheets](https://www.drupal.org/node/171209) defined in the parent theme will be inherited.

**Overriding inherited style sheets:** Specify a style sheet with the same filename in the sub-theme. For instance, to override style.css inherited from a parent theme, add the following line to your sub-theme's .info file:

stylesheets[all][]   = style.css

You will also need to create the style.css stylesheet file; if you simply wish to disable the imported styles, you can create an empty file.

**JavaScript inheritance**

All [JavaScripts](https://www.drupal.org/node/171213) defined in the parent theme will be inherited.

**Overriding inherited JavaScript:** Specify a JavaScript file with the same filename in the sub-theme's .info file. For instance, to override script.js inherited from a parent theme, add the following line to your sub-theme's .info file:

scripts[] = script.js

You will also need to create the script.js file; if you simply wish to disable the imported scripts, you can create an empty file.

**To set Toggle Display of Theme**

**Overriding or adding features:** Specify a new feature in sub-theme's .info file. Add the following lines to your sub-theme's .info file:

features[] = logo  
features[] = name  
features[] = slogan  
features[] = node\_user\_picture  
features[] = comment\_user\_picture  
features[] = comment\_user\_verification  
features[] = favicon  
features[] = main\_menu  
features[] = secondary\_menu

For more detail click here: [Theme settings](https://www.drupal.org/node/221905)

**Template.php function inheritance**

Anything defined in the parent theme's template.php file will be inherited. This includes[theme function overrides](https://www.drupal.org/node/173880#function-override), [preprocess functions](https://www.drupal.org/node/223430) and anything else in that file. Each sub-theme should also have its own template.php file, where you can add additional functions or override functions from the parent theme.

There are two main types of functions in template.php: [theme function overrides](https://www.drupal.org/node/173880#function-override) and[preprocess functions](https://www.drupal.org/node/223430). The template system handles these two types in very different ways.

Theme functions are called through theme('[hook]', $var, ...). When a sub-theme overrides a theme function, no other version of that theme function is called.

On the other hand, preprocess functions are called before processing a .tpl file. For instance, [theme]\_preprocess\_page is called before page.tpl.php is rendered. Unlike theme functions, *preprocess functions are not overridden in a sub-theme*. Instead, the parent theme preprocess function will be called first, and the sub-theme preprocess function will be called next.

There is no way to prevent all functions in the parent theme from being inherited. As stated above, it is possible to override parent theme functions. However, the only way to remove a parent theme's preprocess function is through hook\_theme\_registry\_alter().

**Page, node, block and other template (.tpl.php) file inheritance**

Drupal provides a large set of files that themes can use to inherit properties. By specifying a particular file name and or structure, this allows the theme to override or inherit a template. For more information on this review [working with template suggestions](http://drupal.org/node/223440)

**Drupal 7** Any .tpl.php files from the parent theme will be inherited. You can add template files with more specificity — for instance, node--blog.tpl.php building on an inherited node.tpl.php.

A single hyphen is still used to separate words: for example, "user-picture.tpl.php" or "node--long-content-type-name.tpl.php", so the double hyphen always indicates a more targeted override of what comes before the "--". See Converting 6.x themes to 7.x for more info.

**Drupal 6**: Any .tpl.php files from the parent theme will be inherited. However, to add template files with more specificity, you must also copy over the more general template file from the parent theme manually. For instance, to add a node-blog.tpl.php template in a sub-theme, you must also copy over node.tpl.php from the parent theme. This bug has been fixed in Drupal 7 but [will not be fixed in Drupal 6](http://drupal.org/node/279573#comment-2736592).

**Overriding inherited .tpl.php templates:** Add a template file with the same name in your sub-theme folder to have it override the template from the parent theme.

**Screen shots, logo and favicon inheritance**

The parent theme's screen shot will be inherited. The parent theme's logo (logo.png/logo.jpg) will **not** be inherited. The parent theme's favicon (favicon.ico) will **not**be inherited.

**Overriding inherited screen shots:** Specify a new image file in your sub-theme's .info file.

**Region inheritance**

Sub-themes do not inherit custom regions from a parent theme. If you are using custom regions, you should copy the region declarations from the parent theme's .info file. Be sure your sub-theme's page.tpl.php file matches the sub-theme's region settings.

**Features inheritance**

In Drupal 6, if you use a set of features other than the full range of defaults, these are not inherited from the base theme. If you are using features beyond the default, you should copy the features declarations from the parent theme's .info file.

**Color inheritance**

[Color.module support](https://www.drupal.org/node/108459) within the color directory is **not** inherited.

**Theme settings inheritance**

Theme settings set via [advanced theme settings](https://www.drupal.org/node/177868)' theme-settings.php are **not** inherited, unless you copy the settings declarations from the parent theme's .info file.

Example: Creating a Garland Subtheme That Utilizes the Color Module

*Last updated March 6, 2015. Created on April 12, 2013.  
Edited by [NonProfit](https://www.drupal.org/u/nonprofit" \o "View user profile.), [hksuineg](https://www.drupal.org/user/2677675" \o "View user profile.), [WormFood](https://www.drupal.org/u/wormfood" \o "View user profile.), [hazemnoor](https://www.drupal.org/u/hazemnoor" \o "View user profile.).*[*Log in to edit this page*](https://www.drupal.org/user?destination=node/1969060)*.*

Let's say you want to use Garland as your base theme, but want to be able to make CSS tweaks to it and want to keep the Color module functionality that is integrated in it by creating a subtheme.

My Garland subtheme is called my\_garland.

**Make Your Subtheme Directory and .info File**

Create your subtheme directory in sites/all/themes/my\_garland.

Create a my\_garland.info file and add the following to it. This is how you let Drupal know your subtheme exists and that it should show it in admin/appearance.

name = My Garland  
description = My subtheme of Garland  
core = 7.x  
base theme = garland  
stylesheets[all][] = style.css   
stylesheets[all][] = my\_garland.css

Notice that we're including style.css yet we will not actually create a style.css file. I will explain below.

**Make Your Subtheme CSS File**

Create a my\_garland.css file. This can be empty until you decide to start adding your custom css which will override or be used in addition to Garland's style.css file.

**Copy Garland's Color and images Directory**

In order to use the Color module, you need to copy over the color and images directory from Garland. Copy themes/garland/color and themes/garland/images to sites/all/themes/my\_garland/color and . There should now be a directory in your subtheme folder that is identical to the one in Garland's.

Now that this is here, when you go to your new themes Settings, you will see the color wheel and the dropdown list of color schemes just like in Garland. Note that you need to make the color changes to your subtheme's settings -- not Garland's. The color scheme is not inherited, so your subtheme cannot know about the colors chosen by the parent.

**So, Why Isn't This Working Yet?**

Yet at this point, if you make changes to your subtheme's color scheme the changes do not show up. No matter what colors you may have chosen for my\_garland or Garland, you just get that same icy blue. If you look in themes/garland/color you can see all those icy blue pngs that are currently being used. And you can see style.css referencing them in the images directory:

/\*\*  
\* Layout  
\*/  
.region-header {  
  min-height: 1em;  
  background: #d2e6f3 url(images/bg-navigation.png) repeat-x 50% 100%;  
}

**Template.php Is The Missing Link**

Here's where template.php comes in. And here's also why it is confusing. Subthemes inherit what is in template.php. And if you look closely at that file in Garland, way down at the bottom there's this:

/\*\*  
\* Override or insert variables into the page template.  
\*/  
function garland\_process\_page(&$vars) {  
    // Hook into color.module  
    if (module\_exists('color')) {  
        \_color\_page\_alter($vars);  
    }  
}

Huh, looks like my subtheme should be all set when it comes to the color module. But, apparently not. (**Anyone with the technical knowledge:** it would be interesting to have an idea of why this is -- or why I am incorrect.)

You need to add your own template.php file to your subtheme directory that will mimic this function -- if you simply tried to copy the template.php file from Garland to your subtheme you will generate an error, since (as stated here: <http://drupal.org/node/225125> ), preprocess functions are not overridden, they are repeated. PHP does not like it when two functions with the same name are defined.

So, to keep things simple, just add the function above but alter its name to reflect that it is in your subtheme by renaming it my\_garland\_preprocess\_page():

function my\_garland\_preprocess\_page(&$vars) {  
    // Hook into color.module  
    if (module\_exists('color')) {  
        \_color\_page\_alter($vars);  
    }  
}

And that should be it! Now, remember to clear your theme registry by clearing your cache by going to admin/config/development/performance or using the magic of drush: drush cc. With Garland and your subtheme enabled, and your subtheme set as the default theme, you should now see changes you make to the color in your subtheme's settings.

**What Is Going On Here? My Hypothesis**

So, what exactly is going on here? Why don't you need to have a style.css in your subtheme folder? Because (as far as I can surmise) my\_garland\_process\_page() changes the path to style.css which will be used AND alters some of the content in it reflecting that the .png images now reside in the same directory as style.css. Compare directories within sites/default/files/color/ to the themes/garland/image directory.

That said, even if you do not have a style.css file in your subtheme directory, you still must add it to your .info file with stylesheets[all][] = style.css. Again (I can only surmise) it would seem that my\_garland\_process\_page() depends on this bit of information in the .info file to know it needs to create a reference to it in the header of a page. (\*don't know is it true, as without copy the style.css, the color cannot change)

HOWEVER, if you now take a look at the source of any of your pages, you will only find style.css buried in minified jQuery. So, what **exactly** is going on here remains obfuscated, from my perspective, for good or ill.