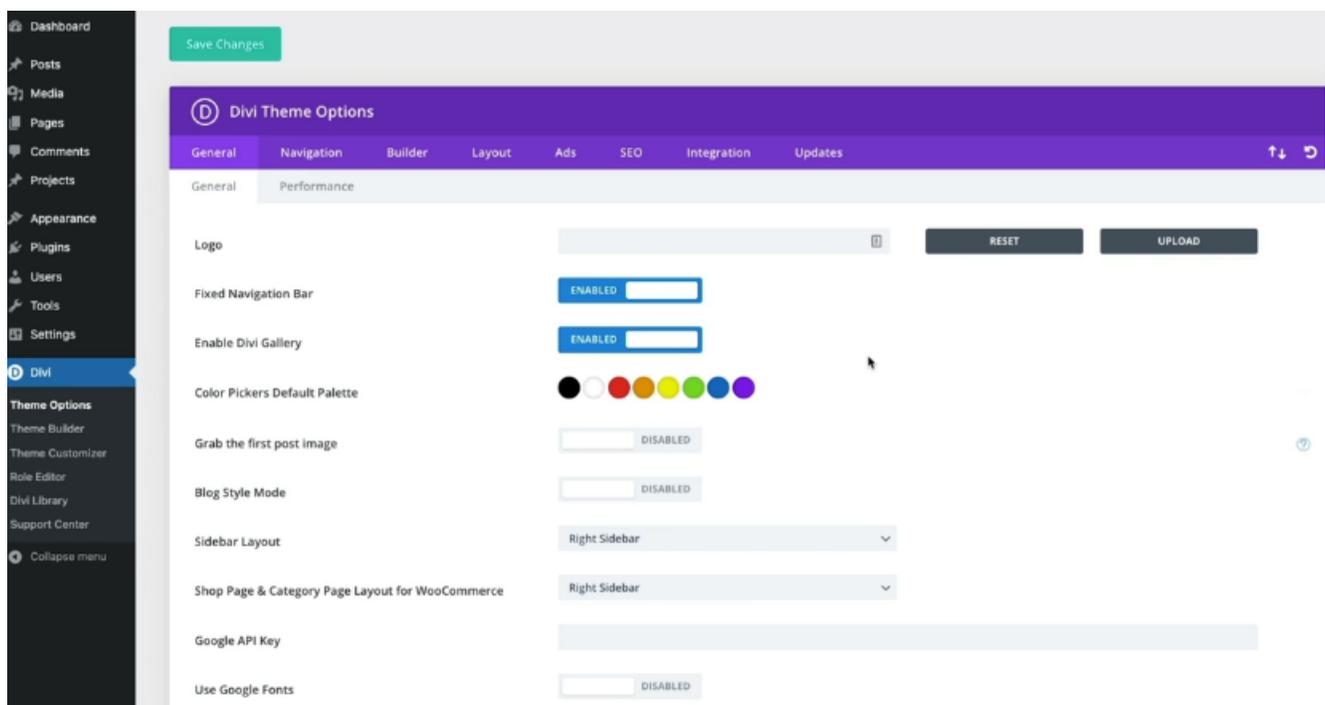# Divi Theme Options

Introduction

Hello and welcome to the first lesson of the Divi Stylist Academy Module 2, where I would like to look into all available Divi Theme options to show you which settings are important and how everything works.

Now, the Divi Theme was created in 2013, so you can imagine a lot of things have changed since then. Elegant Themes - the Divi developers - needed to make sure that

every new update, every new feature works well along with the older features, which some websites may still be using. Especially with version 3.0 when the Visual Builder was introduced, and version 4.0, the release of the Theme Builder. This means Divi became more powerful but also a bit more complicated in terms of finding and using specific features.

All the different settings may be obvious if you've been using Divi for years, but for someone who has been using Divi for a shorter time, maybe even a year or two, some of the options may not be entirely clear.

Now, let me give you an overview of the Divi Theme Options.

Theme Options Settings

And I would like to start from the right side. Theme Options settings can be reset using the round icon here and also moved to a different site using the portability feature with the Export and Import functionality.
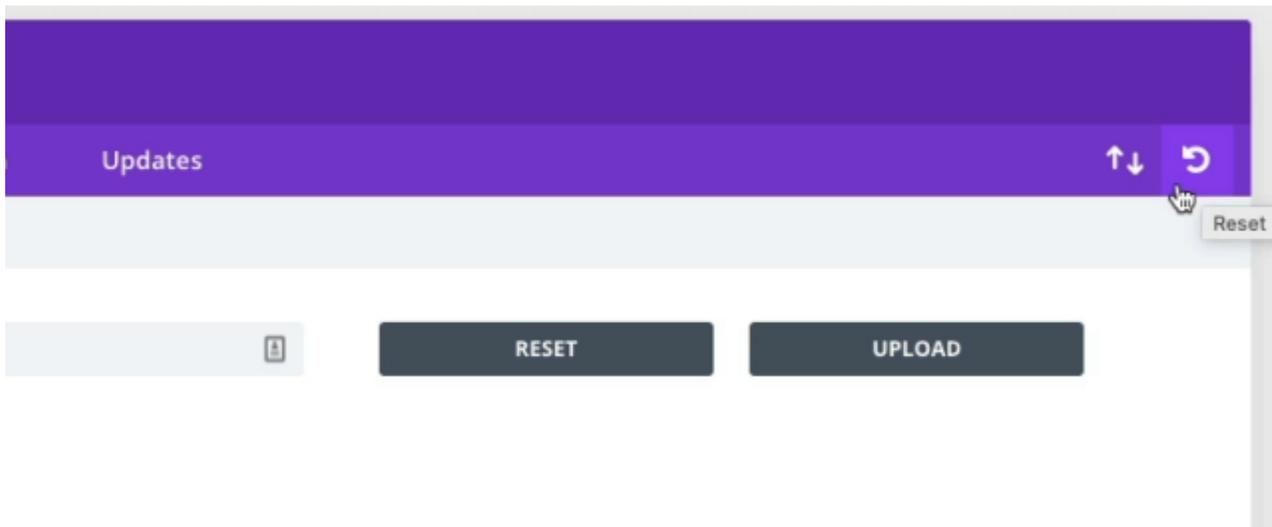
Updates tab

The Updates tab should be the first one you visit after installing the theme. Adding a username and API key is of course necessary to receive theme updates. Also, every time you update Divi to a new version, you can revert this update and roll back to the previous version using the button here.
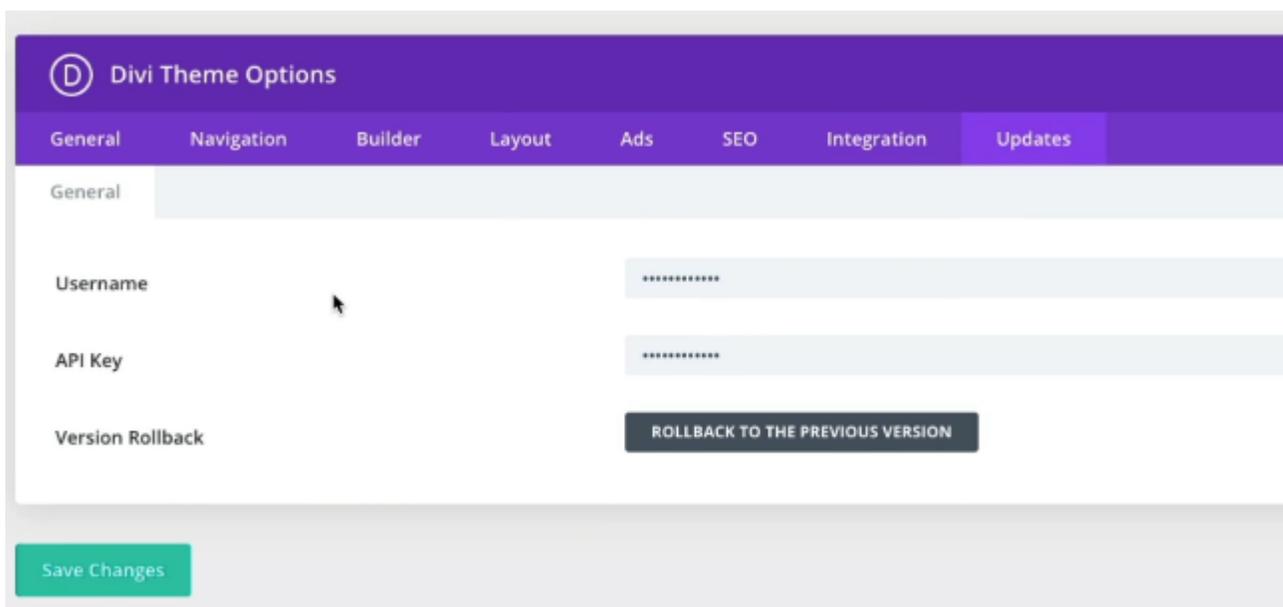
Integration tab

Divi offers four different places where you can add custom code: inside the head tag of the website, inside the body tag to the top of your blog posts, and to the bottom of your posts before the comments. Basically, the Integration tab is a great way to quickly add a snippet of code like Analytics or Facebook pixel.
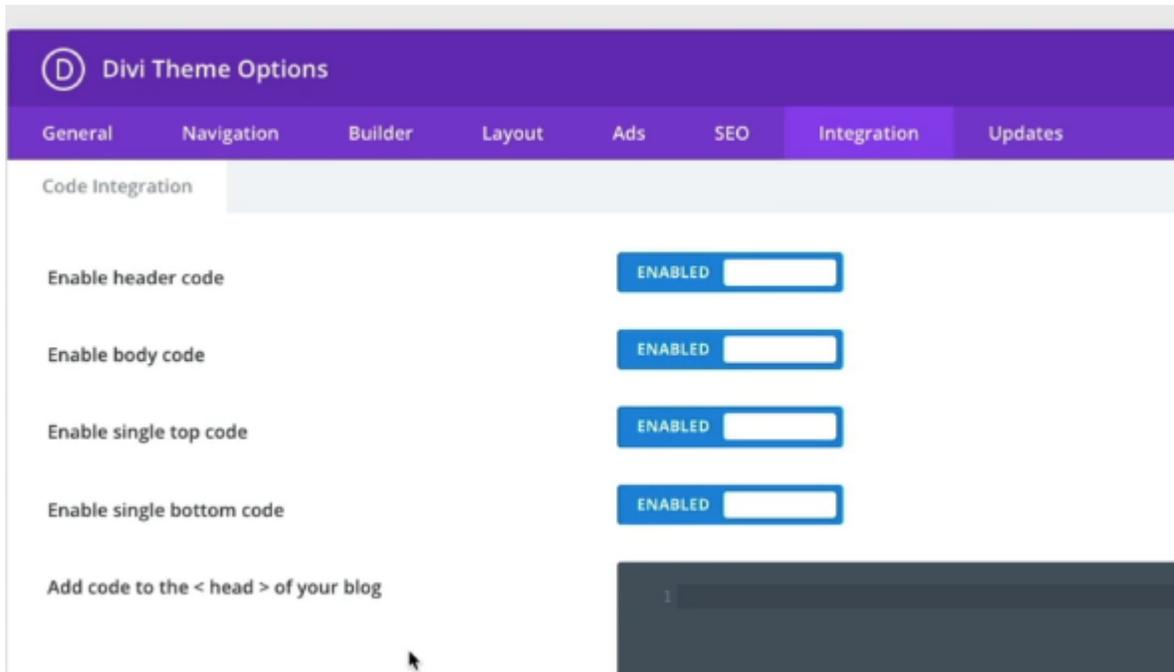

SEO tab

If I'm being honest, I only once had the need to use these settings here. It was for a very small website, one page where I wanted to change the title of the homepage. I think the SEO-specific plugin will always be a better choice and usually, I use one only to make sure I can define additional options like per-post social sharing image, for example.
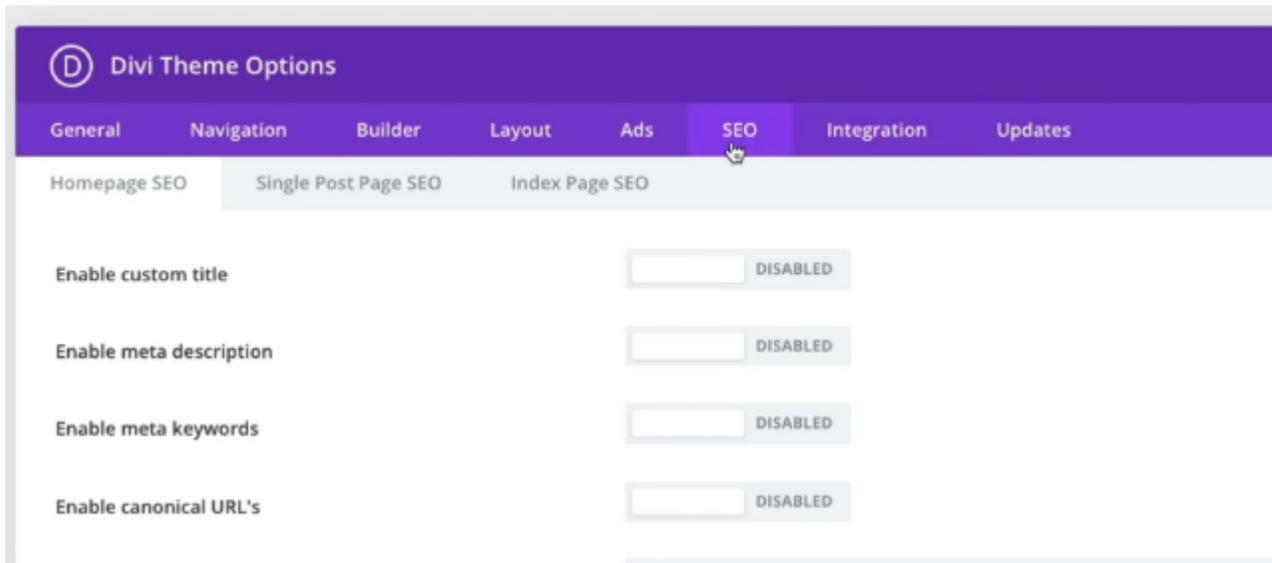
Just to show you, this is how that would work for a single post or pages if you would like to enable custom titles, for example. Once enabled, you can look further down and see the name of the custom field that needs to be used: seo_title.
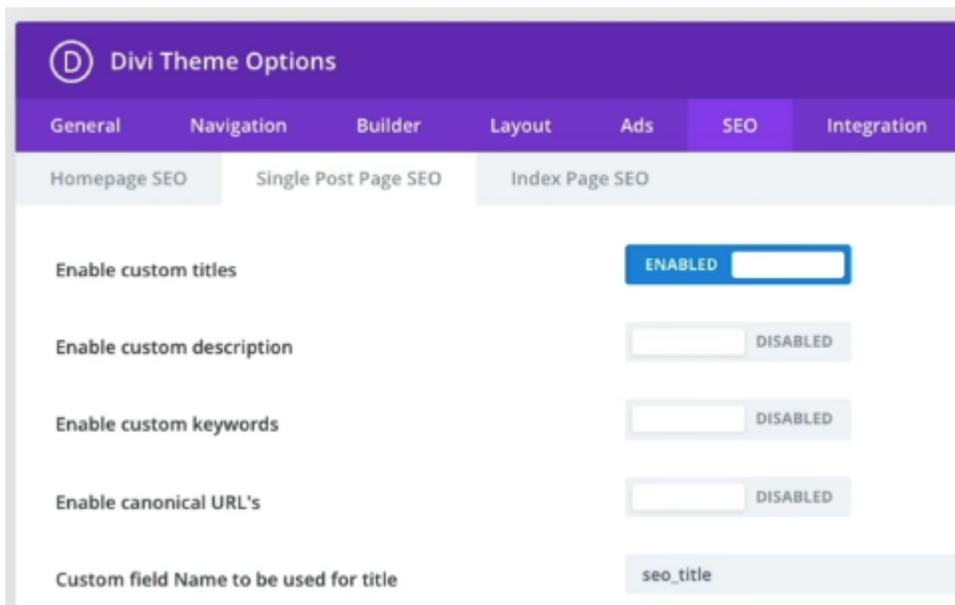
You need it to make it work and to set the new custom title for a blog post or a page. And when I say custom, I mean something that is not just the post title plus website name.

You would need to go to **edit post or page** and make sure **the custom fields are enabled** in the **Preferences**.
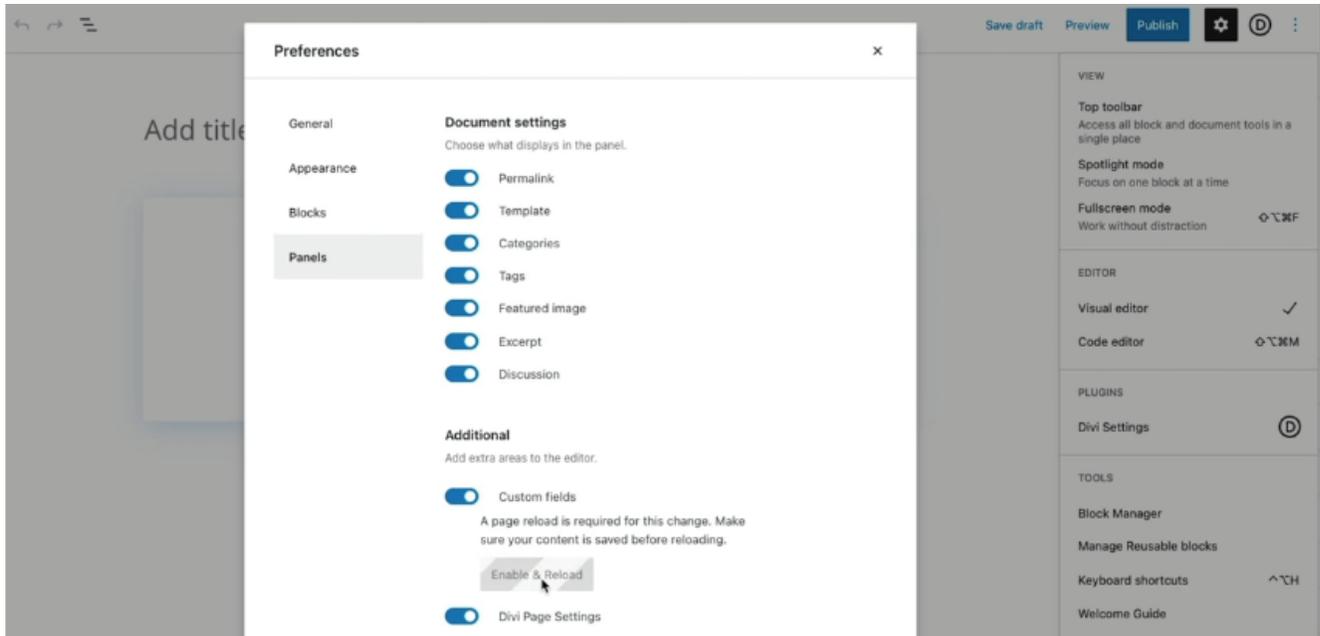
And then add an SEO title custom field with a value that should be visible as the SEO title for that page.

As you can see, there are probably better and more client-friendly ways to do that, so I do recommend using an SEO plugin for that purpose.
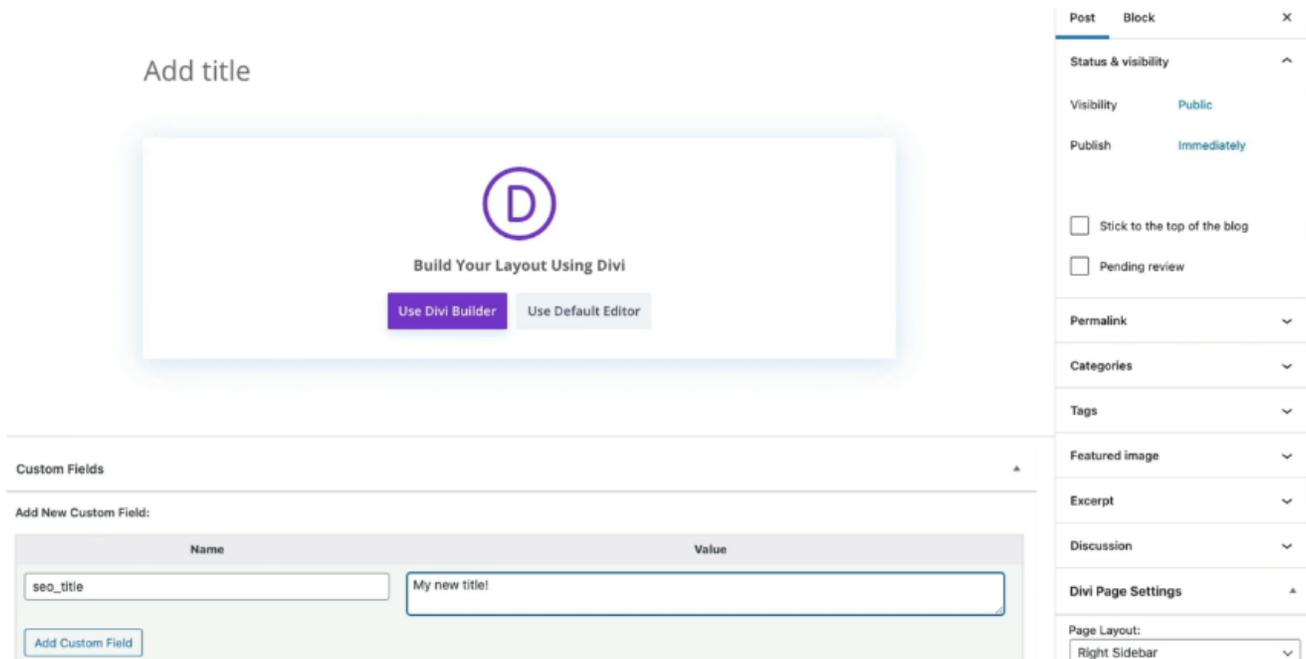
Ads tab

The Ads tab is similarly useful in my opinion. I think that if you want to display ads on your blog, there are better ways to do so, so let's skip this one.

The Layout tab is a bit more interesting.

Divi settings vs Theme Builder settings

But let me clarify something here - because I will be referring to the **Default Divi settings** and **Theme Builder settings** a lot. There is a distinction here, because before Divi 4.0, there were a lot of areas we couldn't easily edit, so some of the settings in Divi Theme Options help with that, but all of these can be overwritten and are no longer relevant if we choose to use the Theme Builder.

Layout tab

For the single post page (which will apply to the default post page if you are not using the Theme Builder to design the content of the post), you can choose which elements are visible in the post meta, here on the single post, and you can decide if you want to display comments, and if you want to show the thumbnail, which is just another name for the featured image.

We have the same options for single pages, except there is no meta for pages. I have never had a situation where I wanted to enable comments on all pages and display a page's featured image. I don't even think I ever assigned a featured image to a page, to be honest. You could use a featured image for pages similar to blog posts on certain websites, but it's probably not the most useful of options.

WordPress Index Pages

The General Settings in the Layout tab refer to the index pages. And let's pause for a moment because I think that is a very important thing to understand about WordPress and Divi.

What is an Index page?

An index page in WordPress is generally any page that displays a list of posts or different post types. So that would include a blog page, a category page, an archive page, tag archives, and so on. When you first install WordPress, you see that index page as your homepage. Then, usually, you navigate to the Settings -> Reading and assign a static page as your homepage.
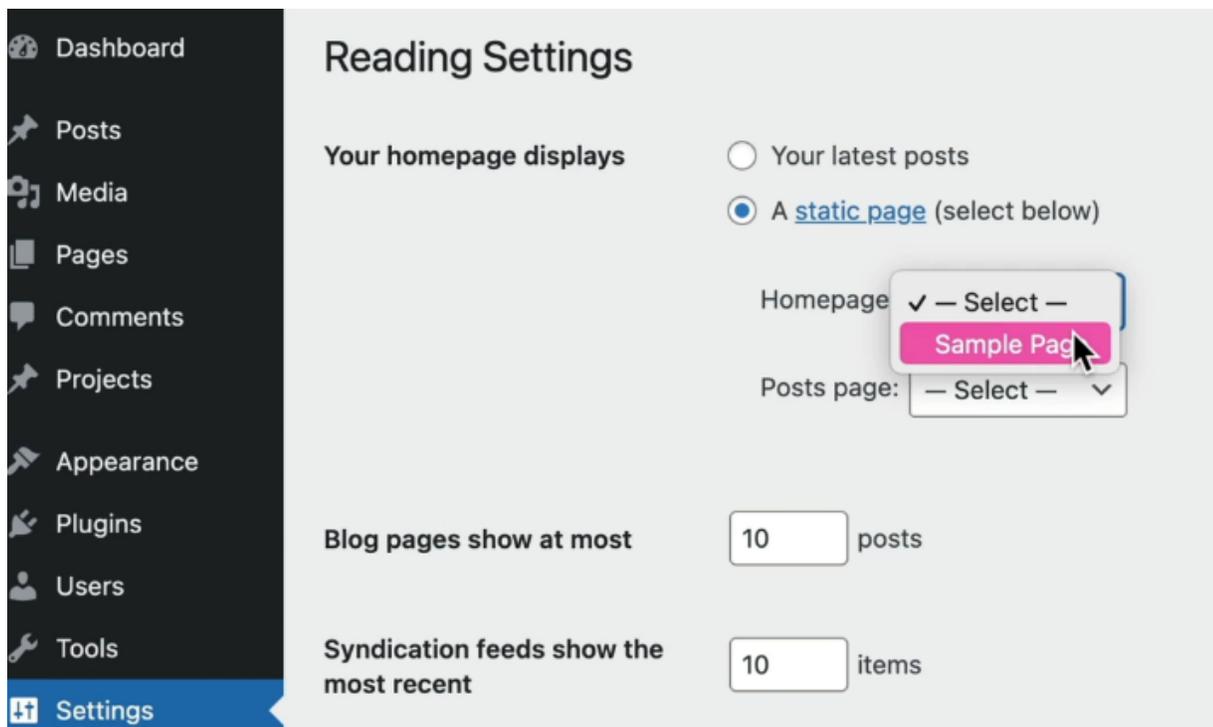
Once a static homepage is selected, you can also assign a **posts page** which is basically your index page. And in Divi that would be called the **blog page**. The blog page is not just a page you add here among other pages with the name "blog", but a blog page is an index page that displays your latest posts. Before the Theme Builder, there was no option to edit that page in Divi without custom code.
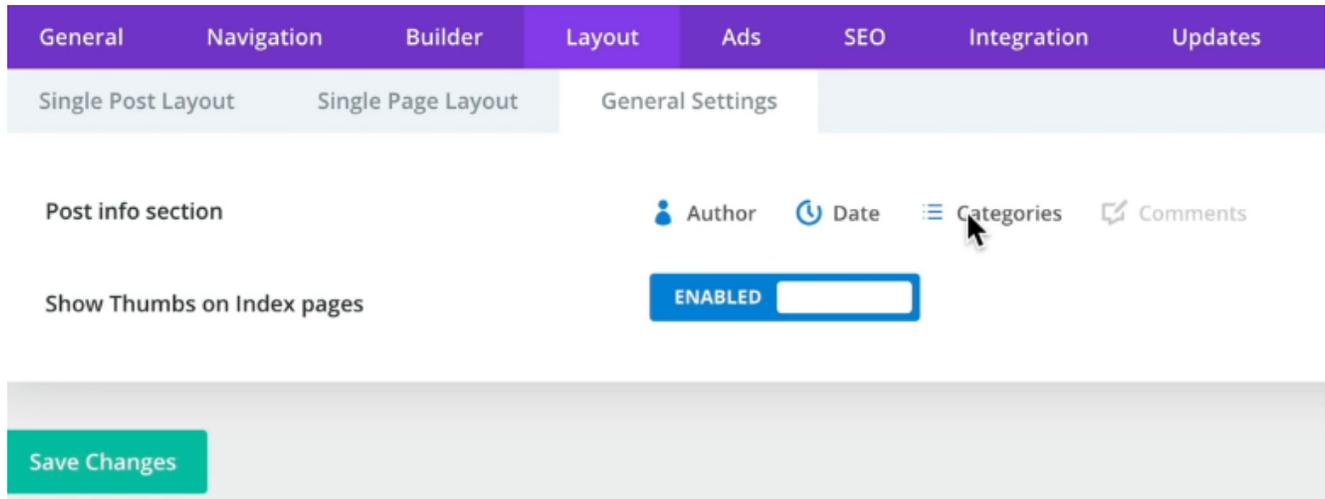
Going back to the General Settings of the Layout tab: here, you can define which elements are visible in the meta of each post on all index pages, and also, if you want to show your featured images on these pages as well. As I mentioned, it all applies to the default index pages, default archives, and categories as long as you do not use the Theme Builder.

Builder tab

In the post type integration, you can choose to disable the builder for certain post types. And I do think it is not a bad idea to disable it for posts. You'll hear comments that it is good not to use Divi for posts, because if you ever decide to switch themes, you'll end up with a bunch of shortcodes.

Divi *Stylist* **Academy** —— **9**
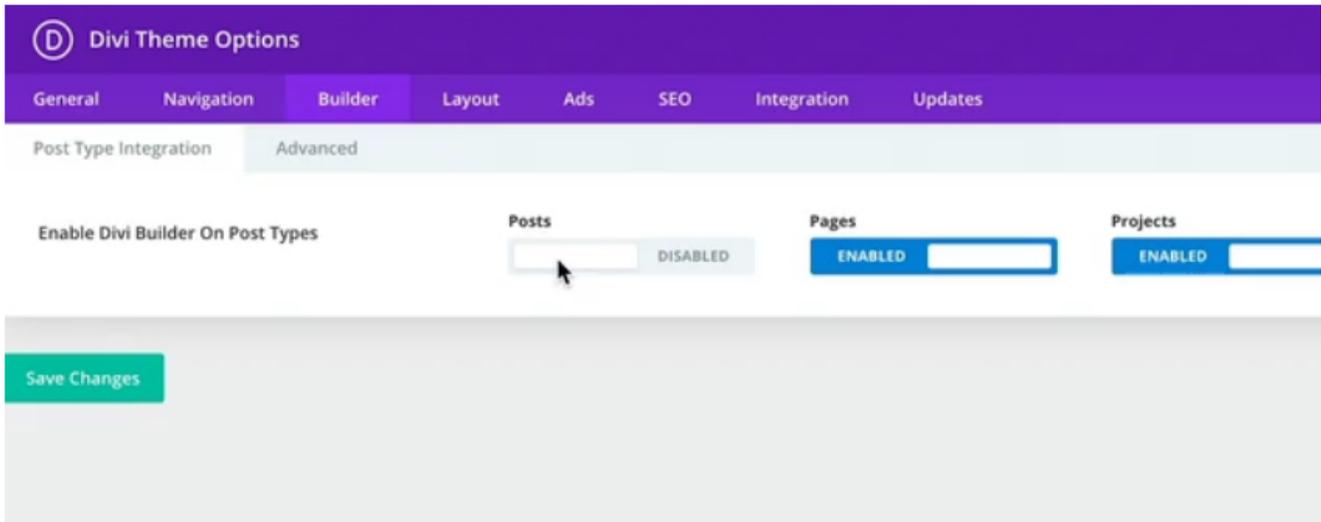
That is not entirely true, and that is also not the reason why I recommend not using Divi Builder for posts. First, I think there are plugins that will strip the shortcodes if needed, and second, you can always use the Divi Builder plugin with a new theme, too.

For me, not using the Builder for posts is a good idea for when you decide to redesign your website. If your posts are heavily customized with the Divi Builder, you may end up having to edit all these posts to match your new website design. Of course, it is your choice, but disabling it here will save you from one additional click in the Post editor, because you are going to be left with just one option - Gutenberg, for example.

Builder - Advanced tab

Speaking of the Post editor, in the Advanced tab of the Builder we can enable the Classic Editor. If you've been using WordPress for a long time, you might be used to the classic version of the page editor but I do think Gutenberg is quite nice, and I think it is important to be familiar with it so I keep this setting disabled.
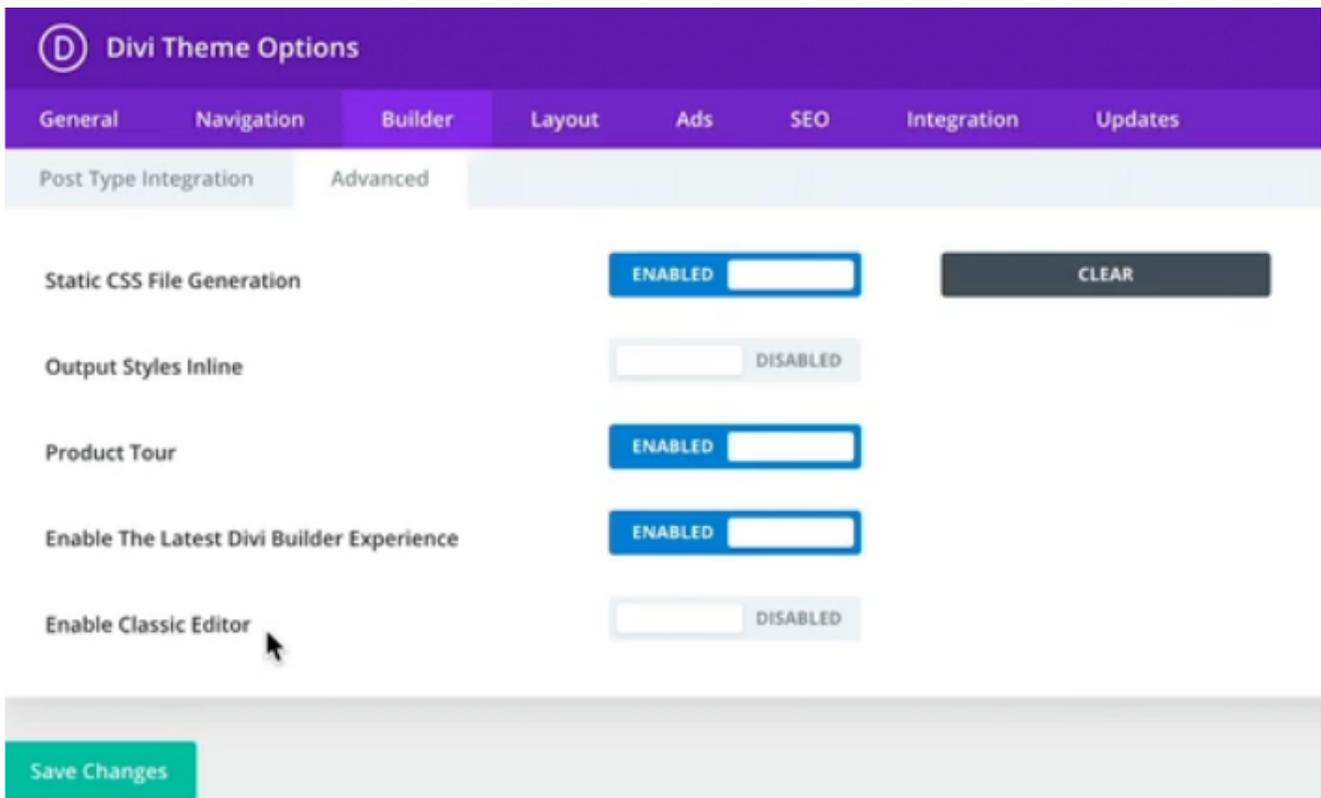
Now for the **Enable The Latest Divi Builder Experience** setting: you want to keep this enabled. It will show the new visual builder while editing Library items, for example. And it is also relevant if the classic builder is enabled, so if you do not want to use the Gutenberg editor, make sure you keep the Latest Builder Experience enabled so that you always use the newest version of the Divi Builder.

Let's look at **Static CSS File Generation**. That is basically a performance improvement that was added to speed the theme up. Instead of making the browser go through the CSS code displayed inline on each page, the CSS files are generated and it speeds the loading time.

However, you have to note that this is the first thing you should check when things do not work or do not look as you think they should.

For example, if you see a different version in the Builder and something else on the frontend, or one of the pages looks different than the others - things like that - the first thing you should do is always to clear the cached CSS file here.

It is not a bad idea to keep this disabled during development to save yourself from any unwanted caching issues.

Navigation tab

Now, here's an interesting tip. With Blog, the settings were no longer relevant once we started using the Theme Builder. Here, these settings are no longer relevant as soon as we create a new menu under Appearance -> Menus.
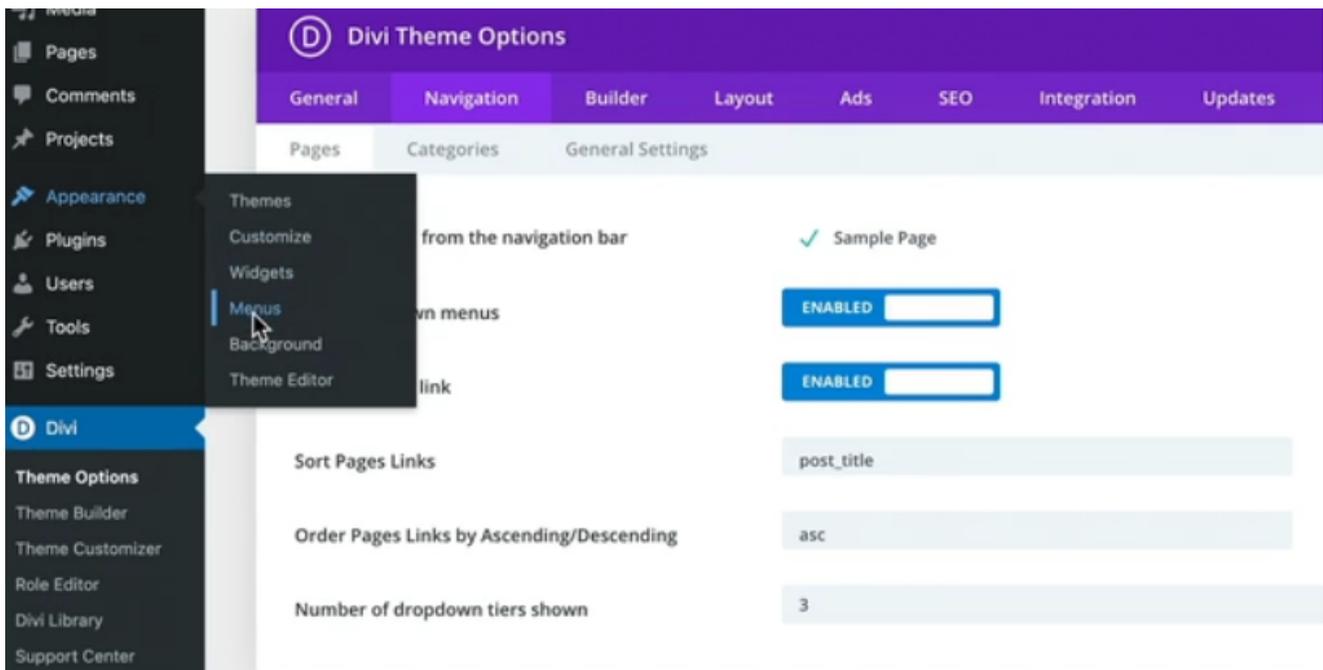
If we do not have any menu created, Divi will generate one for us and add each new page and post categories to the menu, so here we have all sorts of settings to change this default menu, but I have never seen a site that uses this. To be honest, you will most likely want to have more control over your menu and you will usually create a menu yourself, so these settings aren't very important.

The exception is in the **General Settings** tab. Here, you can disable the top-level links if there is a drop down, so that the parent menu item doesn't go anywhere. It will apply to your custom menu too, not just the default one. So if you would like that kind of behavior, here's the place to enable that.
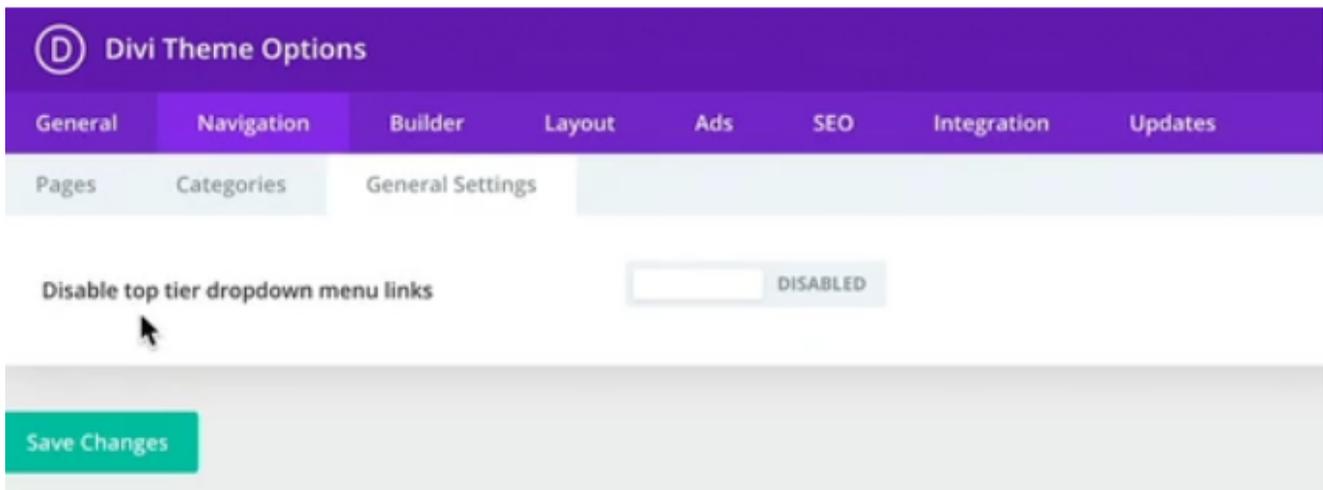
General - Performance tab

Finally, the General tab. Let's have a look at the Performance tab first.

Dynamic Module Framework: it's a dynamic PHP framework. Meaning, if your page only uses 5 of Divi's 50+ modules, Divi only processes the PHP functions needed for those 5 modules and it skips the rest. If those 5 modules don't use some of Divi's many features like sticky options, scroll effects etc., it doesn't process these PHP functions either.

And the same logic applies to the Divi theme stylesheet. With the **Dynamic CSS** option, Divi will generate a small chunk of CSS needed only for the modules you are actually using, instead of loading the full CSS styling and the full stylesheet which targets all of the modules, navigation types, basically all of the theme styling in all of its versions. Dynamic CSS will only load the parts which are actually targeting the elements you are using on your site.
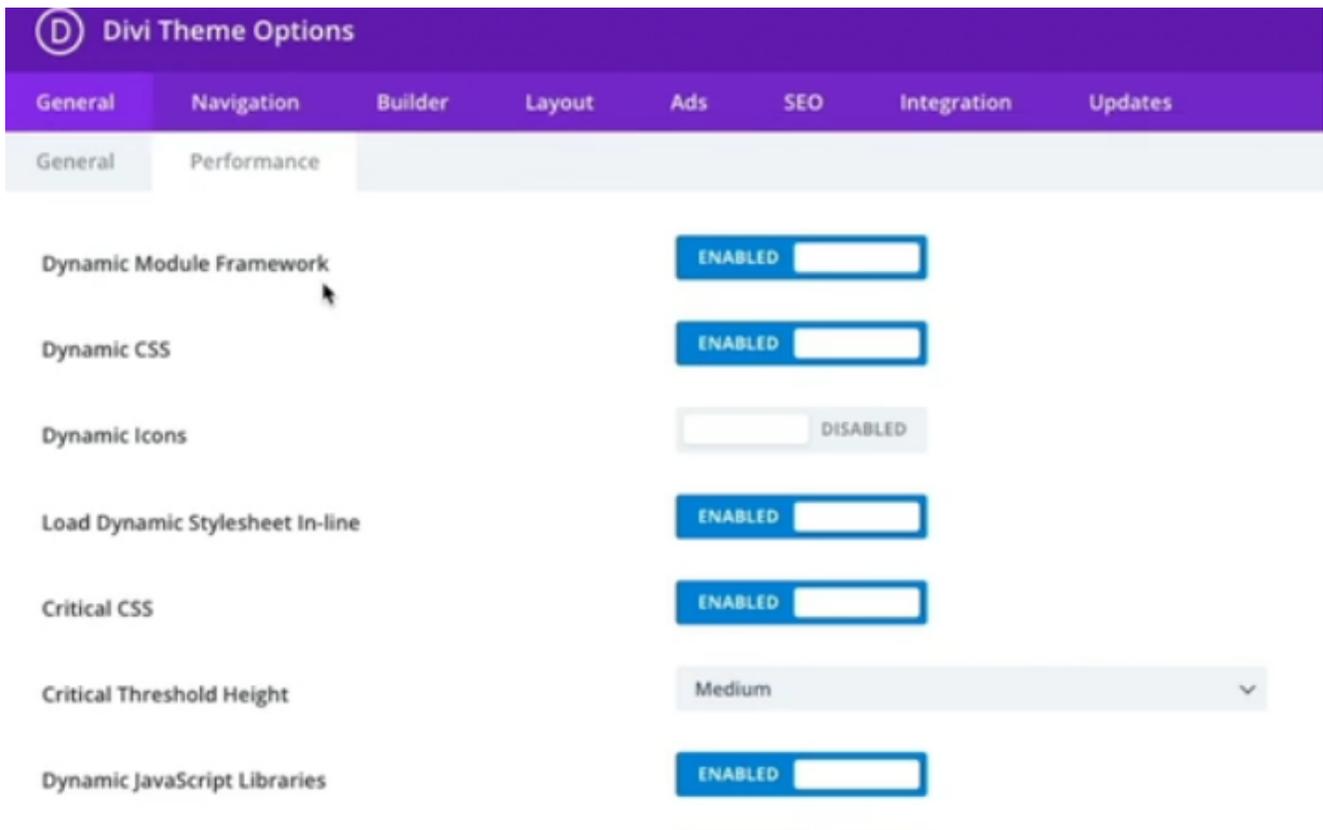
And when the **Dynamic CSS** is enabled, you have these two settings: **Dynamic Icons** and **Load Dynamic Stylesheet In-line**. If the **Dynamic Icons** are disabled, Divi will load the full ETmodules icon font, which you might be using with your custom CSS. If the Dynamic Icons are enabled, the customized, smaller Icon Font version is generated, and it includes only the icons you are using (inside your buttons, blurbs, social follow modules etc). If you want to use your Divi Icons with custom CSS, you have to keep Dynamic Icons disabled.

Enabling **Load Dynamic Stylesheet In-line** improves the PageSpeed scores of individual pages, because it removes this one HTTP request (the render blocking request) by displaying the CSS inside the <style> tag directly in the page source. Placing CSS code on a page this way prevents the style.css file from being cached, but since the stylesheet is small, it's recommended to keep this option enabled.

Now, the **Critical CSS** feature. Critical CSS doesn't load the entire page's stylesheet all at once. Rather, it's very selective with it. It shows what needs to be shown at first sight and loads the rest after the initial interaction. But it's good to remember that if you are not using the fullscreen section as your first element, Divi will automatically detect what appears above the fold on a section level. This means that if you're building a large section that appears partly above the fold and partly below the fold, the entire section's CSS, and the elements' CSS in that section will be loaded. Splitting the bigger sections up into multiple ones will help you avoid having too many elements being treated like they're above the fold.

Related to Critical CSS is the **Critical Threshold Height** option. This threshold is just an estimate and can vary on different devices. Increasing it to High will increase the length of the content considered to be above the fold, deferring (loading later) fewer styles, which results in slightly slower load times. But it's something to consider if you're experiencing CLS - the unexpected shifting of webpage elements while the page is still downloading.

Dynamic JavaScript Libraries: this setting only loads third-party JS libraries if they are actually needed. With this disabled, Divi will for example load a magnific-popup JavaScript library on every page, even though it is only used in the Gallery lightbox. By enabling this option, we are making sure that no unnecessary JS is loaded if our page doesn't need it.

Disable WordPress Emojis: removing WordPress emojis removes unneeded assets and improves performance. And WordPress emojis are not really needed, because modern browsers will show us emojis without loading additional assets.

Defer Gutenberg Block CSS: if a page is built with the Divi Builder, it moves the Gutenberg block CSS file from the header to the footer, which removes another request and improves overall performance.
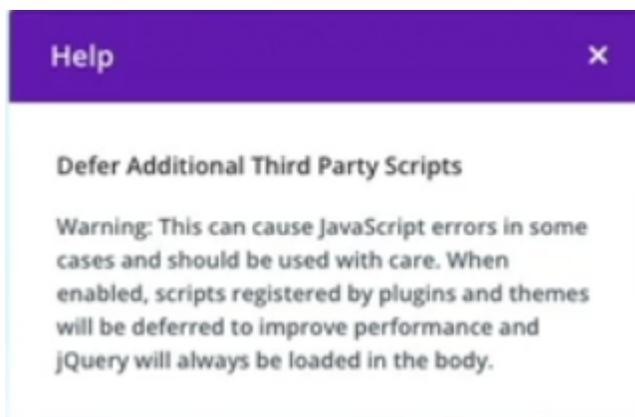
Improve Google Fonts Loading works by enabling caching of Google Fonts and loading them in-line in the header.

Limit Google Fonts Support for Legacy Browsers: enabling this option will lower the size of Google Fonts and improve load times, however, it will limit Google Fonts support in some very old browsers.

Defer jQuery and jQuery Migrate: if the jQuery script is loaded early (in the header), it will pause the loading of the rest of your HTML until the script is executed. This can slow down the rendering of your page, so this setting moves these scripts to the body, but only if there are no other dependencies. If a third-party plugin registers jQuery as a dependency, it will be moved back to the head to avoid conflict. This is something you would need to test, so if you're experiencing troubles with some custom jQuery scripts, you might need to disable this option.

Enqueue jQuery Compatibility Script is another way Divi developers were trying to help us avoid conflicts. If you are using custom jQuery code and it didn't register jQuery as a dependency, this script will help to solve that problem.

Defer Additional Third Party Scripts: this last setting, when enabled, will try to defer (move to the footer) any third party scripts, and it will load jQuery in the body.

Help ✕

Defer Additional Third Party Scripts

Warning: This can cause JavaScript errors in some cases and should be used with care. When enabled, scripts registered by plugins and themes will be deferred to improve performance and jQuery will always be loaded in the body.

That option comes with a warning message: that it can cause JavaScript errors in some cases and should be used with care, so it's just something we need to test ourselves to make use of that additional performance boost.

So these are all of the performance-related Divi settings, which is quite a long list of improvements.

General tab

And lastly, let's see the General tab of the General tab.
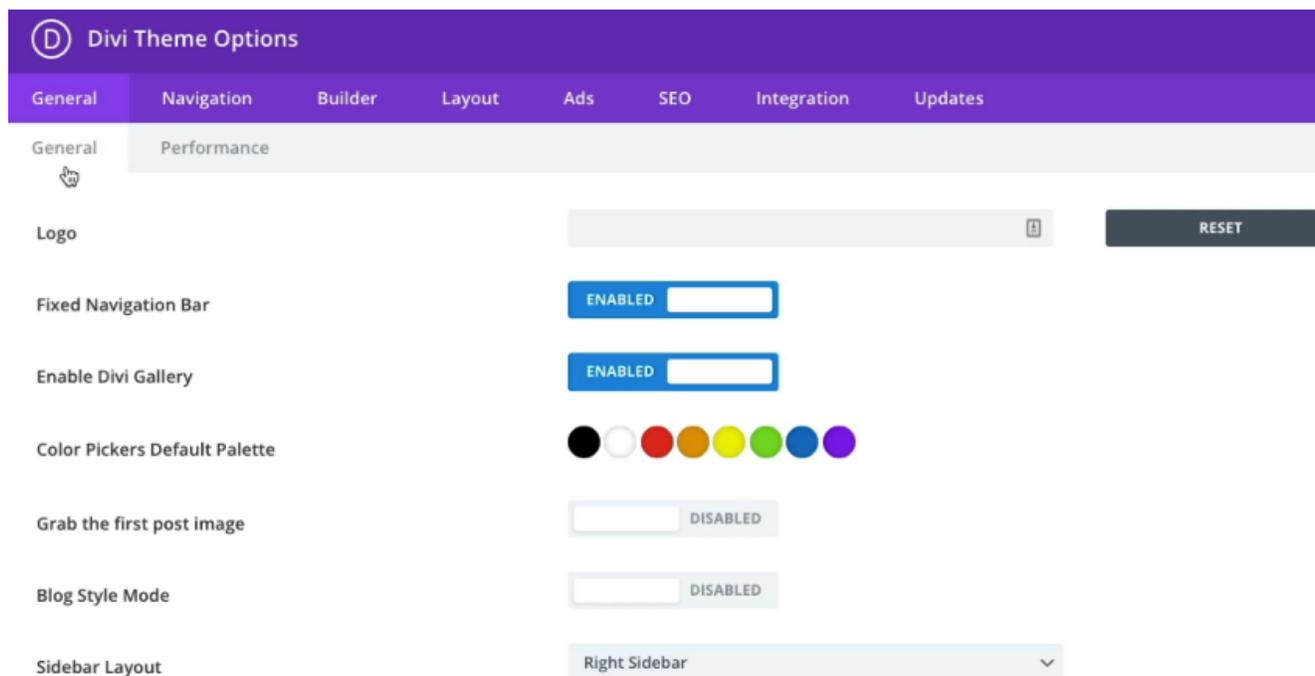
And we have the following settings here: **Logo** and **Fixed Navigation Bar**: two settings only used for the default navigation if you are not using the Theme Builder.

Enable Divi Gallery: this will change the appearance of a gallery that was added through the classic editor, so not very useful anymore.

Color Pickers Default Palette: I would still edit it for a new project but I find that using global colors is more powerful, especially if there's ever a website redesigned down the road, being able to edit all colors in one click is very powerful. And also, because with Global colors you can add more color variations than these eight colors here.

Grab the first post image: this setting is not very common but it will use the first image from the post content as the thumbnail if there is no featured image set.

Blog Style Mode: this setting will display the whole post content on your index pages. Each blog post, the whole post, then you scroll down and there is another post below with the whole content: this is what your category page would look like, for example.

Sidebar Layout: again, it's only useful if you are not using the Theme Builder but is, unfortunately, lacking the option to enable full width layout for single posts. Similarly,

the option **Shop Page & Category Page Layout for WooCommerce** allows you to change the layout for WooCommerce products.

**Google API Key** is required if you are going to use the Map module.

Use Google Fonts and **Google Fonts Subsets:** here you can choose if Google Fonts link calls for a latin-extended subset or not. If you're creating a non-English website and your font needs to support additional characters, you can enable this here OR you can disable Google Fonts completely if you are uploading your own fonts or loading Google Fonts locally.

Enqueue Google Maps Script: this setting can be disabled if you are not using Google Maps within the Map module.

Options for social icons: here, these social icons and the RSS icon can be enabled in the default navigation and the default footer.

Number of posts displayed in certain areas can also be overwritten with the Theme Builder templates.

The **Date format** used in blog meta: you can define your own here, and there are some examples in the WordPress General settings if you'd like to change that.
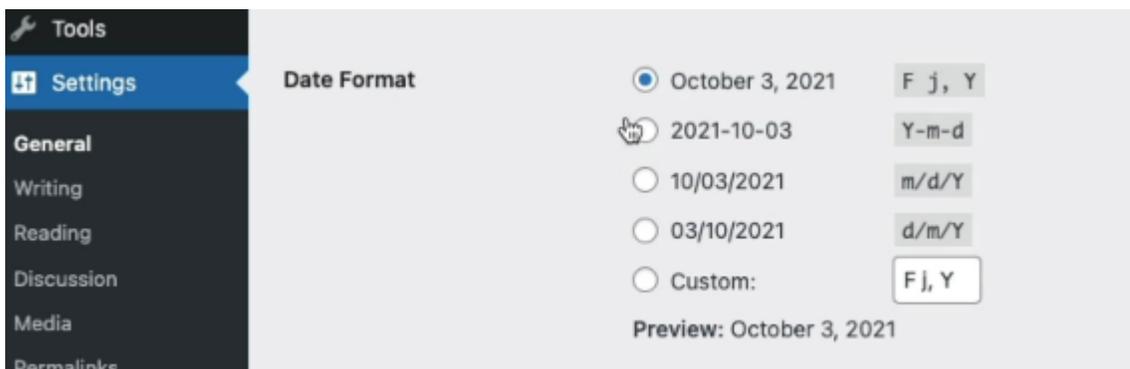
Use excerpts when defined: WordPress excerpt is a text that can be defined for each blog post which then can be displayed on the index pages as the post preview instead of the beginning of the post content. This also applies to default index pages, because with the Theme Builder we can use the Blog module and change this setting using the Blog module options.

Back To Top Button is pretty self-explanatory, and unfortunately, we don't have any customization options for it, but I am going to show you how to edit it with CSS later.

Disable Translations: if your main WordPress language is different than English, you can disable translations on the frontend and for the Divi Builder using this option here.

Enable Responsive Images: by using this feature, we are basically changing the HTML structure of the outputted image tags, which, if enabled, will include the "srcset" attribute to give a browser different image sizes for different screen width.

Finally, my favorite box: **Custom CSS**. This is one of the places you can add custom CSS in Divi. And we'll talk more about that in future lessons. That sums up this little tour.

Resources

USEFUL LINKS:

Local by Flywheel

Action Items

☐ Set up a dummy test Divi website for you to practice and click through all of the settings. You can use Local by Flywheel to create a local WordPress installation and create as many new website versions as you need.