Divi Child Themes

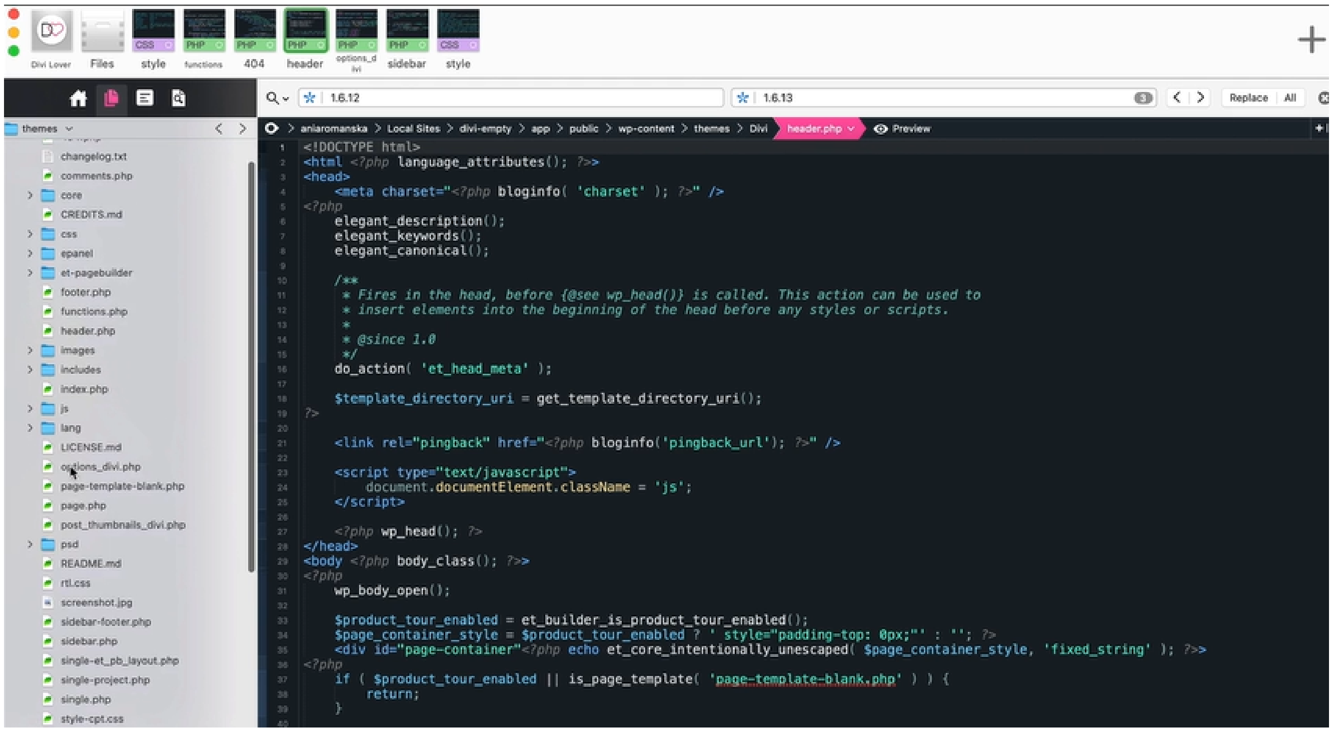
Introduction

Welcome to the lesson about Divi child themes. There might be a lot of different things you might have heard or read about child themes. Some developers say you always need to use one, others say that you shouldn't, so in this video, I would like to explain the confusion and discuss what exactly a WordPress child theme is, how to create one, and when and why you might want to use it. So let's start from the beginning.

What is a WordPress child theme?

Now, I do recommend visiting **the WordPress Codex** and reading about child themes - that is always a great place to learn, and child themes are in no way specific to Divi. This is a WordPress functionality, and on the Codex page, you can find out that "a child theme is a theme that inherits the functionality and styling of another theme called the parent theme". So before learning what a child theme is, we need to understand how the parent themes work.

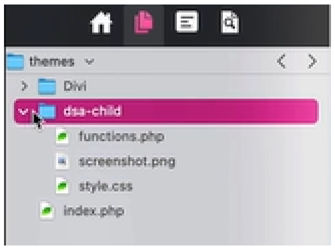
Any WordPress theme such as Divi, or a theme you would get off of **Themeforest**, or a custom-made theme will usually contain a lot of files - mainly PHP files, CSS files, and JavaScript files. All these files are providing the functionality and are defining the appearance of a website that uses the theme.



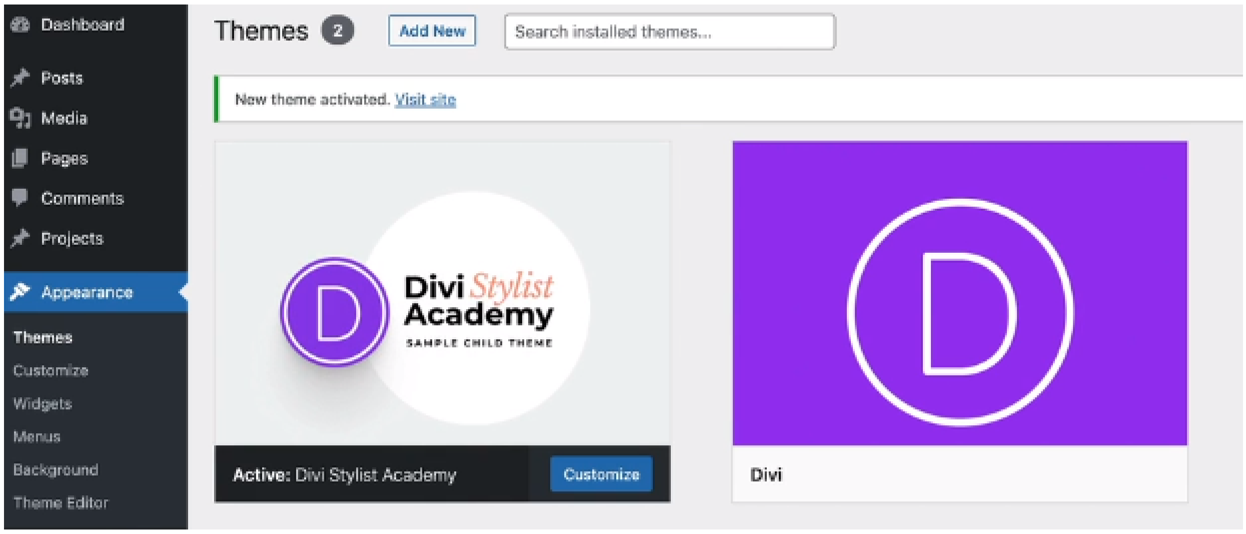
```
<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
	<meta charset="<?php bloginfo( 'charset' ); ?>" />
<?php
	elegant_description();
	elegant_keywords();
	elegant_canonical();

	/**
	 * Fires in the head, before {@see wp_head()} is called. This action can be used to
	 * insert elements into the beginning of the head before any styles or scripts.
	 *
	 * @since 1.0
	 */
	do_action( 'et_head_meta' );

	$template_directory_uri = get_template_directory_uri();
?>

	<link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />

	<script type="text/javascript">
		document.documentElement.className = 'js';
	</script>

	<?php wp_head(); ?>
</head>
<body <?php body_class(); ?>>
<?php
	wp_body_open();

	$product_tour_enabled = et_builder_is_product_tour_enabled();
	$page_container_style = $product_tour_enabled ? ' style="padding-top: 0px;"' : ''; ?>
	<div id="page-container"<?php echo et_core_intentionally_unescaped( $page_container_style, 'fixed_string' ); ?>>
<?php
	if ( $product_tour_enabled || is_page_template( 'page-template-blank.php' ) ) {
		return;
	}
```

Now, you can see all the code in the theme files and make edits and additions, but you should never edit the main theme files! That is, unless you never want to update the theme again. Because a theme update basically replaces all the theme files with the new ones, so updating the theme will overwrite any changes or customizations you made.

And to work around this, WordPress allows us to use a child theme - a separate website layer, where you can place any custom CSS, PHP, or other custom code, or edit any of the parent theme files.



I will explain how exactly we can create a child theme in a moment, but just so you understand the concept: when you create a child theme, and you define that it is a child of a particular parent theme, then all the styling and all the functionality of that parent theme will be automatically used by the child theme. And the child theme would be the active theme in your WordPress panel.



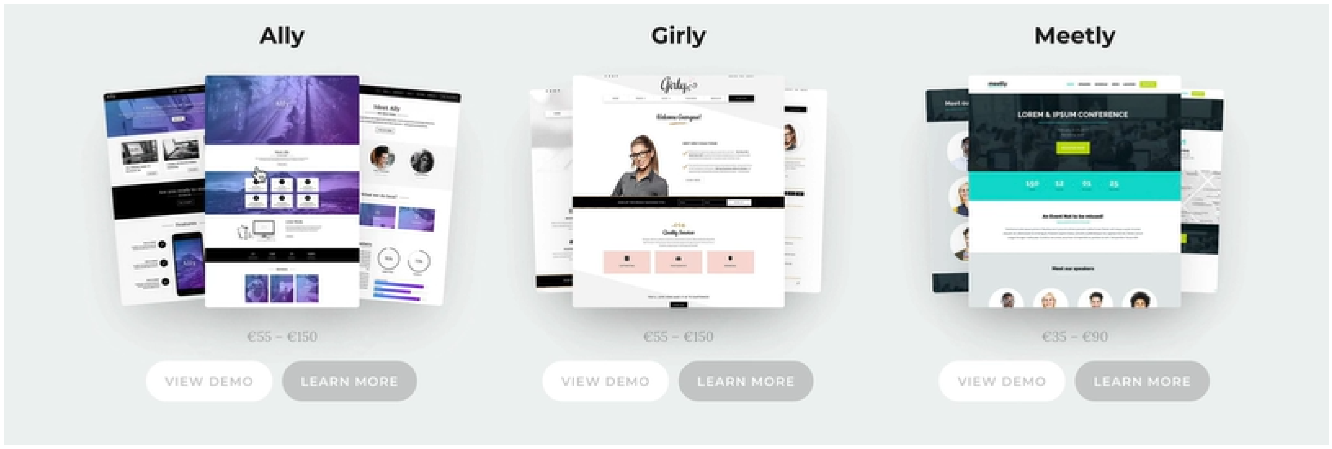
Meaning, WordPress will look into the child theme folder first, and if there is a file in the child theme with the same name as the file in the parent theme - then WordPress will use the child theme's version of the file instead of the parent theme's version.

And this allows us to modify the parent theme files simply by copying the file from the parent theme to your child theme folder and making your edits there.

Blank and premium Divi child themes

Now, when discussing child themes in Divi, I need to mention that there is an additional distinction, and basically two types of child themes you might be thinking of. So we have what is considered a "blank child theme" - the standard WordPress child theme that is created to strictly customize the parent theme. But we also have the concept of "premium child themes", which are used for different reasons despite the same name, as "Divi child theme" used to refer to both.

So what exactly is the difference? A premium child theme is similar to a blank child team in that it sits as a top layer of the parent theme. However, premium Divi child themes are usually fully designed website templates that are created and sold to be used as a starting point for a new website project.



They typically also have additional files with different customizations and allow you to import sample website content demo data, which you can use as a starting point instead of creating all your website pages from scratch. So the goal is to speed up your development time.

Do you need a child theme?

Now, I know many developers say that you should always use a child theme, but it's not entirely true. Using a child theme won't hurt, but you don't always need one.

Divi basically gives you full control over the appearance of your site, but before Divi and other page builders, themes were not as flexible - and that was when you did need a child theme, even for basic customization. There aren't that many situations now where you have the need to edit the PHP files in the Divi Theme these days.

And if you are not planning to add a lot of custom CSS or JavaScript, the chances are you don't need a child theme, because even if you do use custom CSS and a few JavaScript snippets, there are different places in Divi where you can add this code. It doesn't necessarily have to go inside the child theme, and we'll talk more about this in the next lesson.

How to create a child theme?

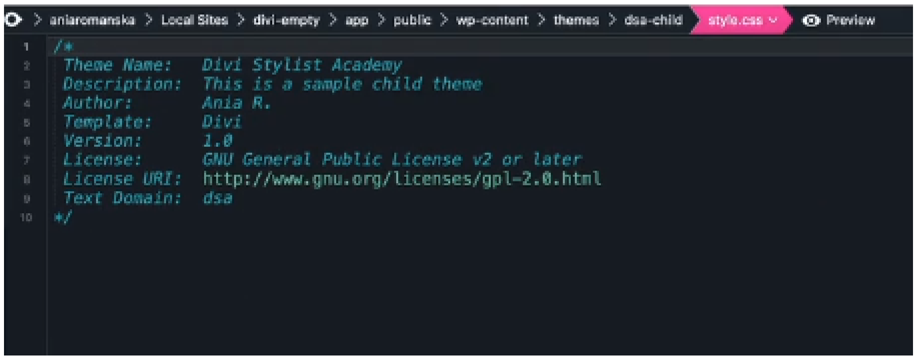
Now, let's look at exactly how you can create a child theme. And you have a few options: you can create one manually, download the sample child theme you'll find in the Downloads section of this lesson, use a child theme generator, or use a plugin.

I want to show you how to build it from scratch, and you can download the child theme we created from the download section of this lesson, and also I am sure a quick Google search will give you a few Divi child theme generators to use (but I haven't tested any, so can't recommend anything specific). There is a plugin I sometimes use - **Child Theme Generator** - but usually, I just do it manually.

So a child theme only really needs two files to work: the style.css file and the functions.php file.

Style.css

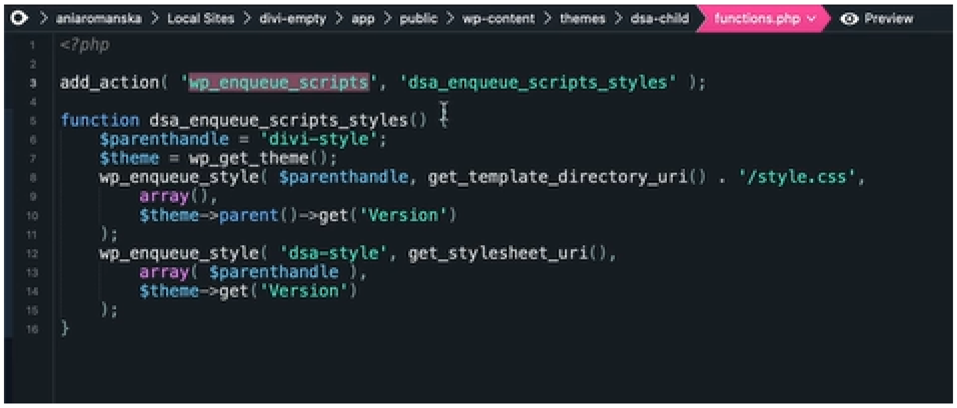
Inside the style.css file, we need to include information about the parent theme so that they are connected, and we can add additional information which will be visible inside the WordPress Appearance page.


aniaromanska > Local Sites > divi-empty > app > public > wp-content > themes > dsa-child > style.css
Preview

```
/*
 Theme Name:   Divi Stylist Academy
 Description:  This is a sample child theme
 Author:       Ania R.
 Template:     Divi
 Version:      1.0
 License:      GNU General Public License v2 or later
 License URI:  http://www.gnu.org/licenses/gpl-2.0.html
 Text Domain:  dsa
*/
```

Functions.php

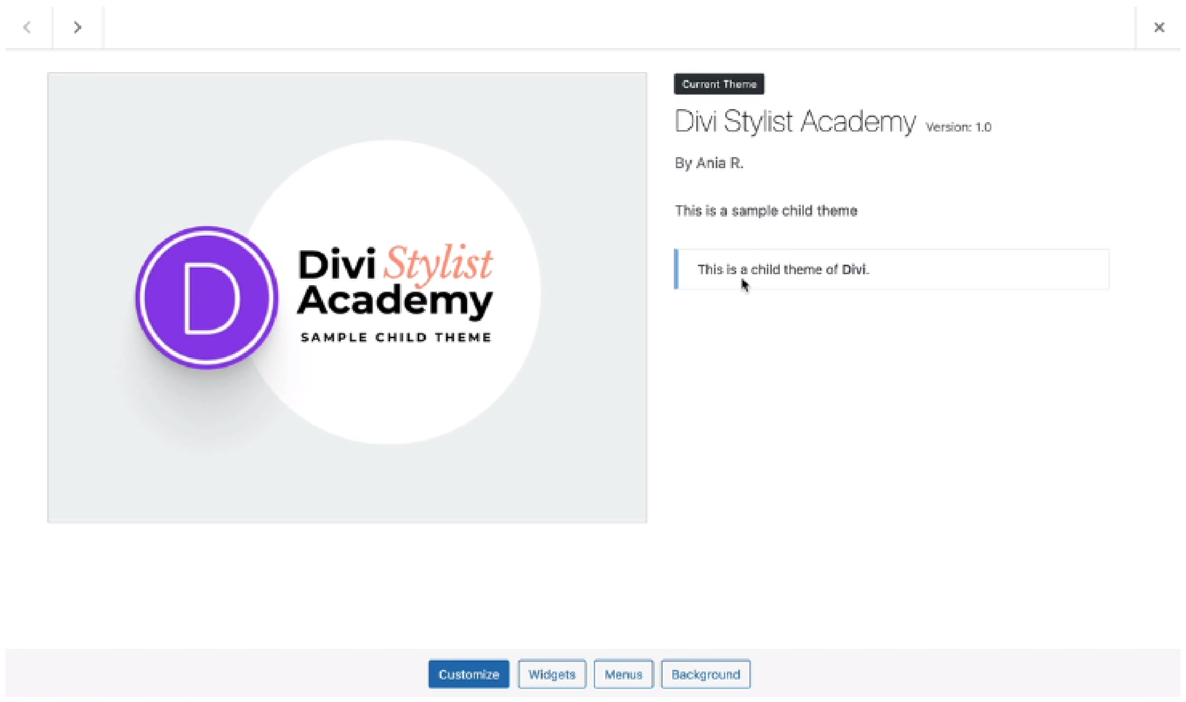
And the functions.php needs to include an enqueue function which loads the parent theme and the child theme's CSS.


aniaromanska > Local Sites > divi-empty > app > public > wp-content > themes > dsa-child > functions.php
Preview

```php
<?php

add_action( 'wp_enqueue_scripts', 'dsa_enqueue_scripts_styles' );

function dsa_enqueue_scripts_styles() {
    $parenthandle = 'divi-style';
    $theme = wp_get_theme();
    wp_enqueue_style( $parenthandle, get_template_directory_uri() . '/style.css',
        array(),
        $theme->parent()->get('Version')
    );
    wp_enqueue_style( 'dsa-style', get_stylesheet_uri(),
        array( $parenthandle ),
        $theme->get('Version')
    );
}
```

Child theme thumbnail

We can also include an additional file: a screenshot.png (or jpeg) to display a child theme thumbnail in the WordPress Dashboard. And, you know, clients love to see their logo here.



Divi Stylist Academy Version: 1.0

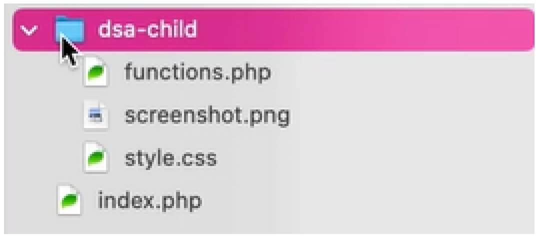
By Ania R.

This is a sample child theme

This is a child theme of Divi.

Upload the child theme

That is basically it. Now, we can use the child theme's style.css file for our custom CSS, and we can include additional files if needed.



And all of the child theme files should be inside a folder which we can upload manually to our website through FTP, or we can zip it and then upload the ZIP file and just simply install the child theme from within WordPress just like any other theme.

So hopefully, this clarifies this topic for you a bit.

Resources

GET INSPIRED:

WP Codex: Child Themes

WP Codex: Conditional Tags

Child Theme Generator plugin

DOWNLOAD A CHILD THEME:

Child Theme Zip File and a Figma Screenshot Template available in the Downloads section of this lesson

Action Items

- ☐ Navigate to the WordPress Codex website and read the article about WordPress child themes.
- ☐ Following the WordPress Codex and looking at the example child theme – create your own blank child theme and try installing it on your test site.