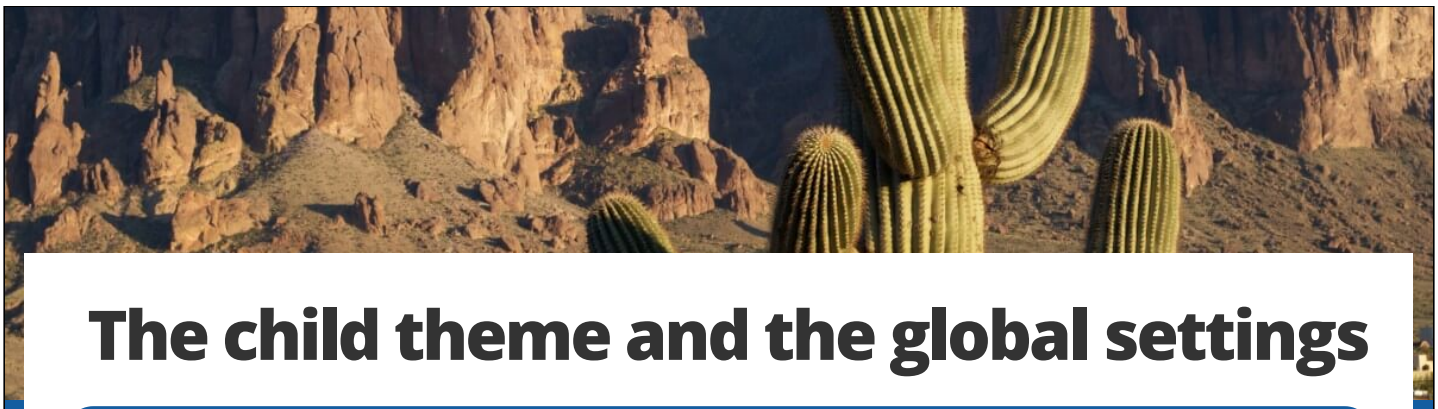


The child theme and the global settings

CSS, Elementor, PHP, Wordpress

Web Skills : 4

The homogeneity of a website is often lacking. The fonts are not identical on all pages or, it is the size. The spacing between the different sections and widgets is another important problem. So, how can you best manage your site, without spending hours on it? Think about the global style sheet

Published Saturday July 20th 2019, 19:24
Modified Saturday July 20th 2019, 19:25



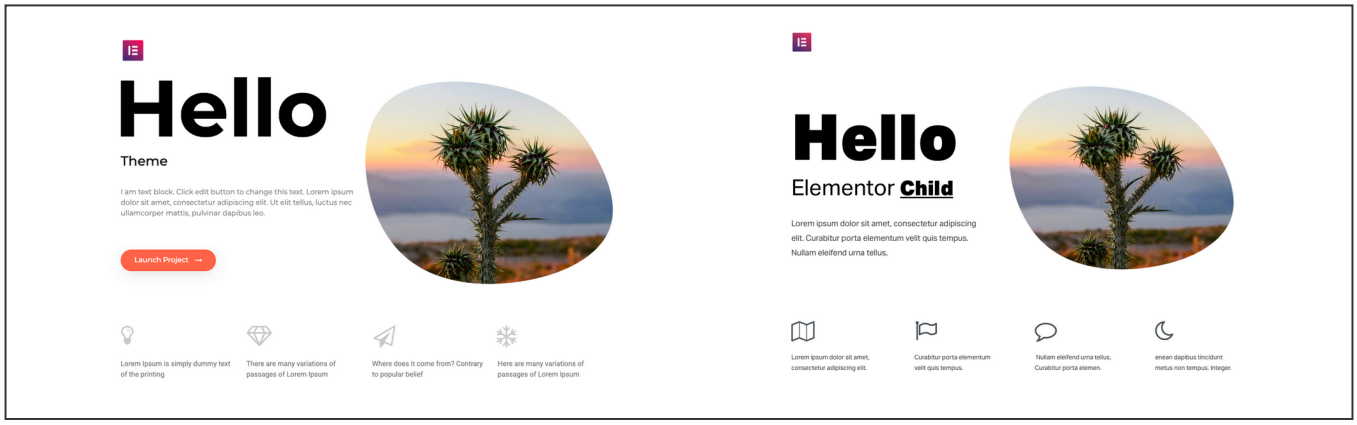
By Olivier Paudex

Introduction

When we reach the limits of the features that WordPress offers as standard, we wonder how some websites do to perform certain manipulations. The answer is quickly given, the **PHP code**. If it can be written using a plugin, it does nothing more than store your code in external files and execute them in a certain order, which is the same as writing it directly in the active theme. It is possible to add PHP code to this chronological order by using actions better known as **hooks**.

The main theme and the child theme

Writing PHP code in your main theme is to be banished once and for all from your habits. Each theme has a file called **"functions.php"**, in which you can freely add your **"hooks"**. Coding a new functionality directly in the "**functions.php"** file is like passing the dust. If it goes away for a few hours, it will come back and the cleaning work will have to be done again. If this is true for the dirt, it is also true for the **PHP** code written in the main theme. If the latter gets updated, all the work done will be lost. But unlike the household chores, WordPress has provided a suitable parade, the child theme. This one is installed in parallel to the main theme and cannot work without it. With Elementor, the recommended theme is of course the one provided by the company itself. It is called **"Hello"**. Its child theme has the same name.



The Hello theme and its child theme

> The **Hello** child theme is attached to this publication. Download it by clicking on the **attachment link** at the top right of this document. It is provided as a ZIP file and installs like any other theme.

A good practice

Rather than writing all the functions in the **"functions.php"** file, it is better to split your own functions in several files, if only for clarity. Below, an extract of **"functions.php"**. Two lines of code (and comments) are added :

The first one will be used to get the folder where the child theme is installed. The second one will indicate that the file **"init.php"** is located at the root of the child theme, in a **"php"** folder.

```
<?php
/**
 * Theme functions and definitions
 *
 * @package HelloElementorChild
 */
// Personnal include
$dir = get_stylesheet_directory();
// Initinclude_once ($dir . '/php/init.php');
```

Once these two lines of codes added, it will be necessary to create the folder **"php"**, then the file **"init.php"** and to save it in the folder **"php"**.

As an example of function, I give you a way to remove the WordPress bar in your site.

```
// Remove admin bar
function remove_admin_bar() {
  if (!is_admin()) {
    show_admin_bar(false);
  }
}add_action('after_setup_theme', 'remove_admin_bar');
```

The style sheets

The style sheets or more commonly called **CSS** for Custom Style Sheet, allow you to set certain objects in a style of your own. To create a style sheet, we start by following the same procedure as for the PHP code. We will add a new PHP file that we will name here **"enqueue_CSS_Styles.php"**.

```
// Init
include_once ($dir . '/php/init.php');include_once ($dir . '/php/enqueue_CSS_Styles.php');
```

Put the style sheet in queue

Once this file is added, we will create it and add these lines of code.

```
<?php
  /**
  * Plugin Name: Enqueue_CSS_Styles
  * Description: Enqueue CSS Styles
  * Author: Olivier Paudex
  * Author Web Site: https://www.fuyens.ch
  */
  // Enqueue CSS Styles
  function Main_Styles() {
    wp_enqueue_style('Main_Styles', get_stylesheet_directory_uri() . '/css/Main_Styles.css', fa
  }
  add_action('wp_enqueue_scripts', 'Main_Styles');
  add_action('admin_enqueue_scripts', 'Main_Styles');?>
```

> Since version 3.0, Elementor has added the ability to manage fonts globally directly in the plugin. You'll tell me that it's much easier than using a child theme and creating a style sheet and you're right. But for all that, style sheets go much further than a few font settings. CSS is the basis of any object transformation in an HTML page.

I let you adapt the header.

This function will allow WordPress to queue a CSS file **"Main_Styles.css"**, which will contain all our global settings. This one will be available to your website when the first page loads. It is the **"wp_enqueue_scripts"** action that will do this. Accessorily, it is also possible to modify the styles of the administration console of WordPress with the action **"admin_enqueue_scripts"**.

Create an H1 style

You will now have to create a **"css"** folder at the root of the theme (in the same place as the**"php"** folder), then create the **"Main_Styles.css"** file. The example below will modify the **H1 style** for the 3 screen formats **(Desktop, Tablet, Smartphone)**.

The values for screen sizes in Elementor are :

- For a mobile, **minimum: 481px, maximum: 1024px.**
- For a tablet, **minimum: 769px, maximum: 1439px.**
- For a desktop computer: **minimum: 1024px.**

> In 2019, with the latest generation mobiles, screens have become very large and tablets are declined in mini and maxi mode, which can be a bit confusing.

Personally, here are the dimensions configured for this site :

- Mobile : **maximum : 619px.**
- Tablet: **minimum: 620px, maximum: 1365px.**
- Desktop computer: **minimum: 1366px.**

If you have copied the code exactly, all your H1 headings will be displayed with the font "Open Sans", a weight of 800, a negative lettering of 1.5px, a line height of 1.3em, centered, and in black.

This will apply to all H1 titles created with the Elementor **"Title"** widget, as well as those created with the **"Text-editor"** widget.

```
/**
* CSS Name: Main_Styles
* Description: Feuille de styles personnalisée
* Author: Olivier Paudex
* Author Web Site: https://www.fuyens.ch
*/
/* Desktop */
@media only screen and (min-width: 1366px) {
  :root {
    --font_size_h1: 42px;
  }
}
/* Tablet */
@media only screen and (min-width: 620px) and (max-width: 1365px) {
  :root {
    --font_size_h1: 38px;
  }
}
/* Smartphone */
@media only screen and (max-width: 619px) {
  :root {
     --font_size_h1: 38px;
  }
}
/* Title H1 */
h1.elementor-heading-title, .elementor-text-editor h1 {
```

```
  font-family: "Open Sans", Sans-Serif;
  font-size: var(--font_size_h1)!important;
  font-weight: 800;
  letter-spacing: -1.5px;
  line-height: 1.3em;
  text-align: center;
  color: #333333;}
```

A good idea, the global settings page

If you want to continue with this approach, I highly recommend that you create a page with all your global settings and add to it all the Elementor widgets whose appearance you want to change globally. Start by adding a paragraph of each type from **H1 to H6** and also of type p and configure your style sheet with your settings. Of course, you need to know a little bit about CSS to achieve your goal.

As an example, I'll show you how to style a button. I open my page where I've already added the H1 title, and I add a button. I don't touch any settings of the button in Elementor. I get this



The settings page with an H1 title and a not yet formatted button

I will add the lines below to my **"Main_Styles.css"** file, and my button will become blue, with no border, rounded corners and the text will be white.

```
/* Button style */
.elementor-widget-button a, button.elementor-button {
  background-color: #2E8BC0;
  color: #FCFCFC!important;
  border-style: none!important;
  border-radius: 5px!important;
  padding: 10px 20px 10px 20px!important;}
```

If I want to center the button, I can add these few lines.

```
/* button center */
.elementor-button-wrapper {
  display: flex;
  justify-content: center;}
```

Here is the result.



If you now add buttons on this page or even on another one, they will all have the same shape. That's the magic of the globalized style sheet.

You'll probably notice in my example that the page contains the domain name, as well as its slogan. This is the **"Hello"** theme that displays this by default. To remove it, you have to create**an header**, which is the subject of the next chapter.