Troubleshooting in WordPress – how to find the culprit?

Tutorial · https://mhthemes.com/support/knb/troubleshooting-wordpress/



Image Source: RyanMcGuire – Pixabay.com / License: CC0 Public Domain

As a beginner (but also as advanced user), your WordPress website can present to you some very challenging and confusing issues from time to time. Overcoming these problems and finding out what's messing up your website can be overwhelming. Especially if you're unfamiliar with modifying code and dealing with scripts. So how to troubleshoot problems on your WordPress site effectively?

- Inspecting your source code
- Deactivating plugins one-by-one
- Troubleshooting implemented scripts
- Checking your applied custom code
- Error reporting via WP Debug Mode

Inspecting your source code

Most popular modern web browsers have an *Inspect Element* tool included. When using the inspection tool of your browser, the browser should expand to show a new area containing the structure of your website (or it may open in a new window). Within this *Inspection Window* you will see the code structure of your website. You can highlight any element with a left-click of your mouse. In turn this will display any CSS code that is currently being applied to the selected element.

> When you're using the *Mozilla Firefox* browser for example, you can install Firebug which is a very useful browser add-on inspection tool to inspect HTML elements on your website and tweak CSS in real-time.

You're also able to collapse and expand every element by clicking the relevant arrow icon. This will bring in to view all other *child-elements* within the element you have just expanded. Using the MH Magazine WordPress theme demo as an example, right-click the menu item *Home* link within the main menu and select your browser's inspection tool.

Locate and select the *mh-main-nav-wrap* element to view the CSS code that is being applied to it. It is highlighted as seen in the image above. That allows you to modify or troubleshoot a particular area. If you're attempting to make CSS modifications to your WordPress website, you can use the inspection tool to find the correct class(es) needed to style your chosen elements.

Modifying code in real-time

By using this method you can quite easily locate elements on your website. You can also modify the CSS styles directly within the inspection window. Simply click on the CSS code you would like to change, enter your new code values and hit *Enter*. Your code changes will be applied in an instant, allowing you to see a live, updated version of your WordPress website within your browser.

These minor tweaks and changes of CSS will persist until you refresh the page you're currently inspecting. Taking advantage of your inspection tool is a great way to visualize how minor style changes will appear on-screen before implementing any actual code changes to your WordPress website. In case you need some help in basic HTML or CSS, please have a look at these tutorials:

- Beginners Guide: HTML Basics for your WordPress Website
- Basic CSS Tutorial to add Custom CSS to your WordPress Website

Deactivating plugins one-by-one

In many cases, when you're having issues on your website, they are related to plugins you might be running. One of the most common solutions to solve these issues on your WordPress website, is to deactivate your plugins. Plugins are developed by a variety of 3rd parties, so on occasion your plugins may conflict with each other, producing seemingly random errors.

If an error code identifies a particular plugin as having an issue, you can deactivate that exact plugin to resolve the problem via your **Plugins** administration menu in your WordPress dashboard. In case that doesn't rectify the issue you can either deactivate all your plugins in a bulk and reactivate them one-by-one. Or you can deactivate only those plugins you suspect to have caused the issue, each at a time to find the problematic plugin, delete it and find an alternative solution.

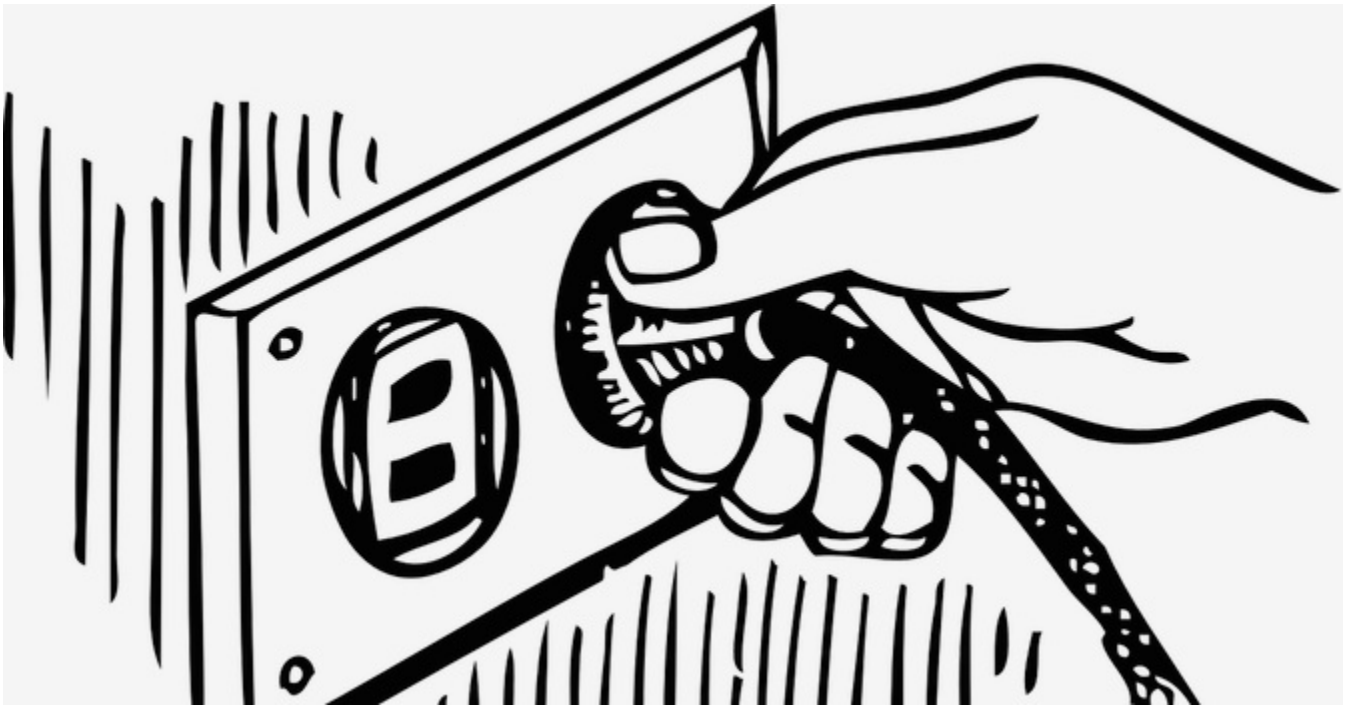


Image Source: Clker-Free-Vector-Images – Pixabay.com / License: CC0 Public Domain

When troubleshooting or dealing with broken plugins, in some cases it might happen that you're being locked out of your WordPress dashboard unable to access anymore. To log back in, you can use your preferred *FTP Client* (like FileZilla) or *File Manager* software instead. Now locate the folder that contains all of your plugins. By default you should find it to be at the following path:

```
/public_html/wp-content/plugins
```

Rename the *plugins* folder to *plugins-old*, as this should now allow you to log back in to your WordPress dashboard. Once you have logged back into your dashboard, rename the *plugins-old* folder **back** to *plugins*. You can now safely access your **Plugins** admin menu in your WordPress dashboard again to troubleshoot all of your plugins as described above.

Troubleshooting implemented scripts

Just as with WordPress plugins you might be running on your site, using multiple *Scripts* from a variety of different sources can also bring incompatibility issues and lead to problems on your WordPress website. When using scripts, ensure they are enqueued right and configured correctly.

Are your scripts being enqueued?

The correct way of implementing and loading scripts into your WordPress website is to *Enqueue* them using a function from within your *functions.php* file. This function will be hooked to

1	wp_enqueue_scripts

in WordPress core. This means, that any script(s) enqueued via this function will be loaded into the correct location for you:

```
if (!function_exists('my_scripts')) {
    function my_scripts() {
        wp_enqueue_script('my-scripts', get_template_directory_uri() . '/js/scripts.js',
array('jquery'));
    }
}
add_action('wp_enqueue_scripts', 'my_scripts');
```

In case a script is being manually loaded by having modified your *header.php* file, it is likely to be out-dated and could be conflicting with other scripts that are correctly being enqueued.

Configuring scripts correctly

With each script you add to your WordPress website, there is a variety of configuration options available for you to use. In some cases they are required for the script to operate correctly. MH Magazine for example, uses a similar function as the one above to load a separate script file that

contains a variety of configuration options and attributes for the various scripts being used.

```
/js/scripts.js
```

To avoid conflicts with other script libraries, all you need within this *scripts.js* file is this code:

```
jQuery(document).ready(function($){
    // script configuration snippets here
});
```

You can now safely add your various script configuration snippets within the above function, using the correct *jQuery* shortcut rather than the standard *$* shortcut. If you require a script to be loaded when the page has loaded you can use this function too:

```
jQuery(window).load(function($){
    // script configuration snippets here
});
```

Script configuration done outside of these functions could conflict with other scripts, or may not even be working at all. To learn more about how to correctly implement jQuery scripts, please see the following article: Beginners Guide: Implement jQuery Scripts into WordPress Websites.

Checking your applied custom code

Modifying WordPress themes should be carried out via the use of a child theme. Your child theme will only contain the files that you have modified. This makes it far easier to manage and it removes the need of adding your custom code after every time you update the *Parent Theme* in the future.



Image Source: ivke32 – Pixabay.com / License: CC0 Public Domain

All of your custom code is safely tucked away within its own child theme folder. This also makes troubleshooting your WordPress website far easier as you don't need to search through the entire theme folder when looking for a solution to your problem - especially when related to custom code.

CSS stylesheet failing to load?

There are some common issues when using child themes, e.g. proper ways to create and use child themes or correct ways to override parent theme functions. Another common issue would be, if the CSS stylesheets for either (or both) parent and child theme are not correctly implemented. You can accomplish this using a single function within your child theme *functions.php* file, like so:

```
function child_styles() {
    wp_enqueue_style('parent-css', get_template_directory_uri() . '/style.css');
    wp_enqueue_style('child-css', get_stylesheet_directory_uri() . '/style.css', array('parent-
css'));
}
add_action('wp_enqueue_scripts', 'child_styles');
```

The parent theme CSS stylesheet file is loaded first. It's then followed by the child theme CSS stylesheet file, which is then associated with the parent CSS file using an array. Notice the difference between how the parent and child theme CSS stylesheets are retrieved?

To locate the parent theme stylesheet file you need to dig slightly deeper outside of the child theme folder using

```
1 get_template_directory_uri()
```

. The location of the child theme stylesheet file is within the child theme folder so you can simply use

```
1 get_stylesheet_directory_uri()
```

. The entire function is then hooked directly to the action responsible for loading all of your WordPress themes scripts and stylesheets on your WordPress website.

Error reporting via WP Debug Mode

Next we'll introduce you to another method to troubleshoot your WordPress website. You can easily take advantage of the WordPress own debug method and enable the Debug Mode for your WordPress website by adding the following line to your *wp-config.php* file:

```
define('WP_DEBUG', true);
```

When you've enabled the Debug Mode, all PHP errors and notices will be visible throughout your entire WordPress website. This allows you to quickly identify problems and look for a suitable solution. However, if you're not familiar with coding, then sometimes these PHP errors may not be very meaningful to you. It's also not recommended to run WP Debug Mode on your live site and in general don't forget to disable debugging when you're done: How to debug your WordPress website.

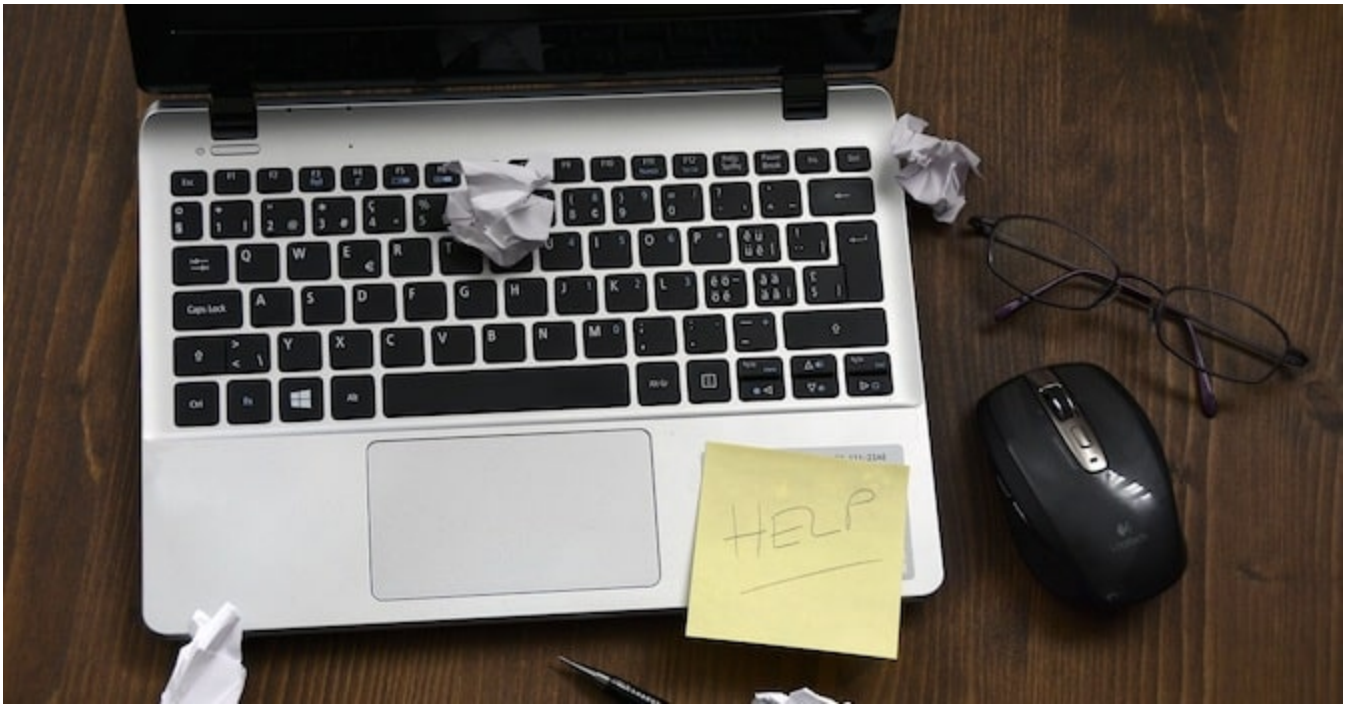


Image Source: annca – Pixabay.com / License: CC0 Public Domain

A good way to find out more about your errors on your WordPress website is to either google it and look for helpful threads online, or to open a thread within the particular WordPress support forums.

Troubleshooting your WordPress website

In this article we've presented a few possible methods to troubleshoot your WordPress website - but for sure there are more. In general, when you're having issues on your website, you should always first check 3rd party plugins, caching, implemented scripts or any custom code you might have applied lately to your website. In case issues appear all of a sudden, then this might also be related to automatic upddates of plugins, changed server configurations by your hosting company or anything else related to your WordPress and / or server environment.