

Debugging



- [Debugging constants](#)
 - [WP_DEBUG](#)
 - [WP_DISABLE_FATAL_ERROR_HANDLER](#)
 - [WP_DEBUG_DISPLAY](#)
 - [WP_DEBUG_LOG](#)
 - [Plugins](#)
-

Debugging is the practice of finding and fixing errors in any software that you build. And it is an essential part of WordPress theme development, regardless of whether you are building a block or classic theme.

WordPress provides some tools out of the box, but there are also plugins that you can use to enhance the experience. You'll learn about some of these methods in this article, but you should also study the [Debugging in WordPress](#) documentation in the Common APIs handbook.

Debugging constants

When creating themes from within a development environment, you should always enable debugging to ensure that your theme is not creating notices or errors. WordPress offers several constants, which you can set in your installation's `wp-config.php` file.

If you open the `wp-config.php` file, scroll down until you locate this code:

```
/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
```

```
* visit the documentation.  
*  
* @link https://wordpress.org/documentation/article/debugging-in-wordpress/  
*/  
define( 'WP_DEBUG', false );
```

This is where you will configure any debugging constants outlined in the sections below.

WP_DEBUG

The `WP_DEBUG` constant is the only one defined in a default WordPress installation's `wp-config.php` file. In a standard install, it is set to `false`, but it is set to `true` if you are running a development copy of WordPress.

The `WP_DEBUG` PHP constant is used to trigger the built-in “debug” mode on your WordPress installation. This allows you to view errors in your theme. It should be used in conjunction with all of the other debugging constants listed in the next sections.

You should see this line in your `wp-config.php` file:

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

To enable debugging, make sure it is set to `true`, as shown here:

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

WP_DISABLE_FATAL_ERROR_HANDLER

WordPress 5.2 [introduced a fatal error handler](#) to ensure that users do not get locked out of their site when a theme or plugin causes a fatal error. This is a great feature in production/live sites. But it can be problematic in development, preventing you from fully diagnosing errors.

For this reason, in development, you should disable this to make sure things are broken until you can fix them. Define this constant in your `wp-config.php` file:

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

WP_DEBUG_DISPLAY

`WP_DEBUG_DISPLAY` is used to control whether debug messages display within the HTML of your WordPress site. By default, when `WP_DEBUG` is enabled, debugging messages will be shown on the screen. So, you can safely not define `WP_DEBUG_DISPLAY` during development.

If you want to disable on-screen debugging messages, you can set `WP_DEBUG_DISPLAY` to `false` in your `wp-config.php` file:

```
define( 'WP_DEBUG_DISPLAY', false );
```

WP_DEBUG_LOG

The `WP_DEBUG_LOG` constant is an optional feature that you can set to log errors to a `debug.log` file in your site's `/wp-content` directory. By default, this is disabled.

Generally, if you turn off on-screen debugging messages via `WP_DEBUG_DISPLAY`, then you will want to opt for storing these messages in the debug log.

To enable logging, define the `WP_DEBUG_LOG` constant as `true` in your `wp-config.php` file:

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

Plugins

There are several plugins that are helpful when debugging your theme. Each is hosted in the WordPress Plugin Directory:

- **[Debug Bar](#)**: Provides a central location for debugging in your WordPress toolbar.
- **[Query Monitor](#)**: Enables debugging of database queries, PHP errors, hooks and actions, block editor blocks, enqueued scripts and stylesheets, HTTP API calls, and more.
- **[Log Deprecated Notices](#)**: Logs incorrect and deprecated function and file usages in your theme.

👤 #3

📅 2026-08-29 23:36:32 UTC

📅 2026-10-31 06:44:07 UTC