

SECOND EDITION



Professional

**WordPress<sup>®</sup>**

**Plugin Development**

Brad Williams, Justin Tadlock, John James Jacoby

# Table of Contents

[COVER](#)

[FOREWORD](#)

[INTRODUCTION](#)

[WHO THIS BOOK IS FOR](#)

[WHAT YOU NEED TO USE THIS BOOK](#)

[WHAT THIS BOOK COVERS](#)

[HOW THIS BOOK IS STRUCTURED](#)

[CONVENTIONS](#)

[SOURCE CODE](#)

[ERRATA](#)

[1 An Introduction to Plugins](#)

[WHAT IS A PLUGIN?](#)

[AVAILABLE PLUGINS](#)

[ADVANTAGES OF PLUGINS](#)

[INSTALLING AND MANAGING PLUGINS](#)

[SUMMARY](#)

[2 Plugin Framework](#)

[REQUIREMENTS FOR PLUGINS](#)

[BEST PRACTICES](#)

[PLUGIN HEADER](#)

[DETERMINING PATHS](#)

[ACTIVATE/DEACTIVATE FUNCTIONS](#)

[UNINSTALL METHODS](#)

[CODING STANDARDS](#)

[SUMMARY](#)

### [3 Dashboard and Settings](#)

[ADDING MENUS AND SUBMENUS](#)

[PLUGIN SETTINGS](#)

[THE OPTIONS API](#)

[THE SETTINGS API](#)

[KEEPING IT CONSISTENT](#)

[SUMMARY](#)

### [4 Security and Performance](#)

[SECURITY OVERVIEW](#)

[USER PERMISSIONS](#)

[NONCES](#)

[DATA VALIDATION AND SANITIZATION](#)

[FORMATTING SQL STATEMENTS](#)

[SECURITY GOOD HABITS](#)

[PERFORMANCE OVERVIEW](#)

[CACHING](#)

[TRANSIENTS](#)

[SUMMARY](#)

### [5 Hooks](#)

[UNDERSTANDING HOOKS](#)

[ACTIONS](#)

[FILTERS](#)

[USING HOOKS FROM WITHIN A CLASS](#)

[USING HOOKS WITH ANONYMOUS FUNCTIONS](#)

[CREATING CUSTOM HOOKS](#)

[FINDING HOOKS](#)

[SUMMARY](#)

### [6 JavaScript](#)

[REGISTERING SCRIPTS](#)  
[ENQUEUEING SCRIPTS](#)  
[LIMITING SCOPE](#)  
[LOCALIZING SCRIPTS](#)  
[INLINE SCRIPTS](#)  
[OVERVIEW OF BUNDLED SCRIPTS](#)  
[POLYFILLS](#)  
[YOUR CUSTOM SCRIPTS](#)  
[jQuery](#)  
[BACKBONE/UNDERSCORE](#)  
[REACT](#)  
[SUMMARY](#)

## [7 Blocks and Gutenberg](#)

[WHAT IS GUTENBERG?](#)  
[TOURING GUTENBERG](#)  
[PRACTICAL EXAMPLES](#)  
[TECHNOLOGY STACK OF GUTENBERG](#)  
[“HELLO WORLD!” BLOCK](#)  
[WP-CLI SCAFFOLDING](#)  
[CREATE-GUTEN-BLOCK TOOLKIT](#)  
[BLOCK DIRECTORY](#)  
[SUMMARY](#)

## [8 Content](#)

[CREATING CUSTOM POST TYPES](#)  
[POST METADATA](#)  
[META BOXES](#)  
[CREATING CUSTOM TAXONOMIES](#)  
[USING CUSTOM TAXONOMIES](#)

## [A POST TYPE, POST METADATA, AND TAXONOMY PLUGIN](#)

### [SUMMARY](#)

## [9 Users and User Data](#)

### [WORKING WITH USERS](#)

### [ROLES AND CAPABILITIES](#)

### [LIMITING ACCESS](#)

### [CUSTOMIZING ROLES](#)

### [SUMMARY](#)

## [10 Scheduled Tasks](#)

### [WHAT IS CRON?](#)

### [SCHEDULING CRON EVENTS](#)

### [TRUE CRON](#)

### [PRACTICAL USE](#)

### [SUMMARY](#)

## [11 Internationalization](#)

### [INTERNATIONALIZATION AND LOCALIZATION](#)

### [CREATING TRANSLATION FILES](#)

### [SUMMARY](#)

## [12 REST API](#)

### [WHAT THE REST API IS](#)

### [WHAT YOU CAN DO WITH THE REST API](#)

### [ACCESSING THE WORDPRESS REST API](#)

### [THE HTTP API](#)

### [WORDPRESS' HTTP FUNCTIONS](#)

### [BRINGING IT ALL TOGETHER](#)

### [SUMMARY](#)

## [13 Multisite](#)

[TERMINOLOGY](#)

[ADVANTAGES OF MULTISITE](#)

[ENABLING MULTISITE IN WORDPRESS](#)

[MULTISITE FUNCTIONS](#)

[DATABASE SCHEMA](#)

[QUERY CLASSES](#)

[OBJECT CLASSES](#)

[SUMMARY](#)

#### [14 The Kitchen Sink](#)

[QUERYING AND DISPLAYING POSTS](#)

[SHORTCODES](#)

[WIDGETS](#)

[DASHBOARD WIDGETS](#)

[REWRITE RULES](#)

[THE HEARTBEAT API](#)

[SUMMARY](#)

#### [15 Debugging](#)

[COMPATIBILITY](#)

[DEBUGGING](#)

[ERROR LOGGING](#)

[QUERY MONITOR](#)

[SUMMARY](#)

#### [16 The Developer Toolbox](#)

[CORE AS REFERENCE](#)

[PLUGIN DEVELOPER HANDBOOK](#)

[CODEX](#)

[TOOL WEBSITES](#)

[COMMUNITY RESOURCES](#)

[TOOLS](#)

[SUMMARY](#)

[INDEX](#)

[END USER LICENSE AGREEMENT](#)

## List of Tables

Chapter 3

[TABLE 3-1: List of Core Sections and Fields](#)

Chapter 12

[TABLE 12-1: Main HTTP Status Codes](#)

[TABLE 12-2: HTTP Status Code Classes](#)

[TABLE 12-3: Default Settings of wp\\_remote\\_  
Functions Optional Parameters](#)

## List of Illustrations

Chapter 1

[FIGURE 1-1: Loading a page in WordPress](#)

[FIGURE 1-2: Plugins menu](#)

[FIGURE 1-3: Install Now button](#)

[FIGURE 1-4: Types and statuses for plugins](#)

Chapter 3

[FIGURE 3-1: Custom registered menu](#)

[FIGURE 3-2: Submenus](#)

[FIGURE 3-3: Submenu labeled PDEV Settings](#)

[FIGURE 3-4: Plugin management page](#)

[FIGURE 3-5: Error message](#)

[FIGURE 3-6: Section appended](#)

[FIGURE 3-7: Singular field](#)

[FIGURE 3-8: Heading levels](#)

[FIGURE 3-9: Dashicons](#)

[FIGURE 3-10: Dismissable notices](#)

[FIGURE 3-11: WordPress-styled button](#)

[FIGURE 3-12: Link styled to look like a button](#)

[FIGURE 3-13: WordPress-like options](#)

[FIGURE 3-14: Table style](#)

[FIGURE 3-15: Pagination style](#)

## Chapter 4

[FIGURE 4-1: Insufficient privileges](#)

[FIGURE 4-2: Expired link message](#)

[FIGURE 4-3: Rogue JavaScript running](#)

[FIGURE 4-4: Related Posts list](#)

## Chapter 7

[FIGURE 7-1: Classic Editor, not covered in this chapter](#)

[FIGURE 7-2: Gutenberg](#)

[FIGURE 7-3: Block Library menu](#)

[FIGURE 7-4: Categories of blocks](#)

[FIGURE 7-5: View options](#)

[FIGURE 7-6: Sidebar's Document menu](#)

[FIGURE 7-7: Sidebar's Block menu and formatting toolbar](#)

[FIGURE 7-8: WooCommerce blocks](#)



[FIGURE 7-9: Newest Products block](#)

[FIGURE 7-10: Event Calendar blocks](#)

[FIGURE 7-11: Post Type Switcher plugin](#)

[FIGURE 7-12: webpack finishing successfully](#)

[FIGURE 7-13: Our “Hello world!” block in the Block Library](#)

[FIGURE 7-14: Selecting our new block](#)

[FIGURE 7-15: Editing a post](#)

[FIGURE 7-16: WP-CLI scaffold generated](#)

[FIGURE 7-17: Build Step 1](#)

[FIGURE 7-18: Build Step 2](#)

[FIGURE 7-19: Build Step 3](#)

[FIGURE 7-20: My Block in Block Library](#)

[FIGURE 7-21: My Block in Content Area](#)

## Chapter 8

[FIGURE 8-1: Books admin menu and screen](#)

[FIGURE 8-2: Tags submenu item](#)

[FIGURE 8-3: Genres submenu](#)

## Chapter 9

[FIGURE 9-1: New form on the user edit page](#)

[FIGURE 9-2: New roles](#)

## Chapter 10

[FIGURE 10-1: Scheduled Events page](#)

[FIGURE 10-2: Number output](#)

## Chapter 11

[FIGURE 11-1: Settings box](#)

## Chapter 12

[FIGURE 12-1: Unformatted JSON](#)

[FIGURE 12-2: Formatted JSON](#)

[FIGURE 12-3: Post results](#)

[FIGURE 12-4: Error message](#)

[FIGURE 12-5: Authentication worked](#)

[FIGURE 12-6: “My Time at Crystal Lake” post](#)

## Chapter 13

[FIGURE 13-1: Tools ⇔ Network menu options](#)

## Chapter 14

[FIGURE 14-1: Simple post list](#)

[FIGURE 14-2: Widgets admin screen](#)

[FIGURE 14-3: Favorites list](#)

[FIGURE 14-4: Custom dashboard widget](#)

[FIGURE 14-5: Custom dashboard widget](#)

## Chapter 15

[FIGURE 15-1: Admin Toolbar menu item](#)

[FIGURE 15-2: Query Monitor interface](#)

[FIGURE 15-3: Database queries](#)

## Chapter 16

[FIGURE 16-1: Function with parameters](#)

[FIGURE 16-2: Codex search options](#)

[FIGURE 16-3: WordPress PHPXref](#)

[FIGURE 16-4: Function list](#)

# **PROFESSIONAL WORDPRESS® PLUGIN DEVELOPMENT**

**Second Edition**

**Brad Williams**

**Justin Tadlock**

**John James Jacoby**



**WROX**<sup>TM</sup>

A Wiley Brand

# FOREWORD

This book will teach you how to develop for WordPress. WordPress has, over the past two decades, grown into the CMS that powers more than one-third of all websites. If you're proficient at WordPress development, you'll never be out of a job again.

Starting out as a simple blogging system, over the last few years WordPress has morphed into a fully featured and widely used content management system. It offers individuals and companies worldwide a free and open source alternative to closed source and often very expensive systems.

When I say fully featured, that's really only true because of the ability to add any functionality needed in the form of a plugin. The core of WordPress is simple: you add in functionality with plugins as you need it. Developing plugins allows you to stand on the shoulders of a giant: you can showcase your specific area of expertise and help users benefit while not having to deal with parts of WordPress you don't care or know about.

When I wrote the foreword of this book's first edition, nine years ago, I'd just started my own company. That company has since grown to consist of 100+ people, and our plugins are used on more than 10 million sites—all through the power of open source and plugins.

I wished that when I started developing plugins for WordPress as a hobby, almost 15 years back, this book had been around. I used it as a reference countless times since, and I still regularly hand this book to new colleagues.

The authors of this book have always been a source of good information and wonderful forces in the WordPress

community. Each of them is an expert in his own right; together they are one of the best teams that could have been gathered to write this book, and I'm glad they're here for a second edition.

WordPress makes it easy for people to have their say through words, sound, and visuals. For those who write code, WordPress allows you to express yourself in code. And it's simple. Anyone can write a WordPress plugin. With this guide in hand, you can write a plugin that is true to WordPress' original vision: code is poetry.

*Happy coding!*

*Joost de Valk*

[Yoast.com](http://Yoast.com)

# INTRODUCTION

Dear reader, thank you for picking up this book! You have probably heard about WordPress already, the most popular self-hosted content management system (CMS) and blogging software in use today. WordPress powers literally millions of websites on the Internet, including high-profile sites such as TechCrunch and multiple Microsoft websites. What makes WordPress so popular is that it's free, open source, and extendable beyond limits. Thanks to a powerful, architecturally sound, and easy-to-use plugin system, you can customize how WordPress works and extend its functionalities. There are already more than 55,000 plugins freely available in the official plugin repository, but they won't suit all your needs or client requests. That's where this book comes in handy!

As of this writing, we (Brad, Justin, and John) have publicly released more than 100 plugins, which have been downloaded millions of times, and that's not counting private client work. This is a precious combined experience that we are going to leverage to teach you how to code your own plugins for WordPress by taking a hands-on approach with practical examples and real-life situations you will encounter with your clients.

The primary reason we wanted to write this book is to create a preeminent resource for WordPress plugin developers. When creating plugins for WordPress, it can be a challenge to find the resources needed in a single place. Many of the online tutorials and guides are outdated and recommend incorrect methods for plugin development. This book is one of the most extensive collections of plugin development information to date and should be considered

required reading for anyone wanting to explore WordPress plugin development from the ground up.



## **WHO THIS BOOK IS FOR**

This book is for professional web developers who want to make WordPress work exactly how they and their clients want. WordPress has already proven an exceptional platform for building any type of site from simple static pages to networks of full-featured communities. Learning how to code plugins will help you get the most out of WordPress and have a cost-effective approach to developing per-client features.

This book is also for the code freelancers who want to broaden their skill portfolio, understand the inner workings of WordPress functionality, and take on WordPress gigs. Since WordPress is the most popular software to code and power websites, it is crucial that you understand how things run under the hood and how you can make the engine work your way. Learning how to code plugins will be a priceless asset to add to your résumé and business card.

Finally, this book is for hobbyist PHP programmers who want to tinker with how their WordPress blog works, discover the infinite potential of lean and flexible source code, and learn how they can interact with the flow of events. The beauty of open source is that it's easy to learn from and easy to give back in turn. This book will help you take your first step into a community that will welcome your creativity and contribution.

Simply put, this book is for anyone who wants to extend the way WordPress works, whether it is for fun or profit.

## **WHAT YOU NEED TO USE THIS BOOK**

This book assumes you already have a web server and WordPress running. For your convenience, it is preferred that your web server runs on your localhost, as it will be

easier to modify plugin files as you read through the book, but an online server is also fine.

Code snippets written in PHP are the backbone of this book. You should be comfortable with reading and writing basic PHP code or referring to PHP's documentation to fill any gaps in knowledge about fundamental functions. Advanced PHP code tricks are explained, so you don't need to be a PHP expert.

You will need to have rudimentary HTML knowledge to fully understand all the code. A basic acquaintance with database and MySQL syntax will help with grasping advanced subjects. To make the most of the chapter dedicated to JavaScript and Ajax, comprehension of JavaScript code will be a plus.

## **WHAT THIS BOOK COVERS**

As of this writing, WordPress 5.5 is around the corner, and this book has been developed alongside this version.

Following the best coding practices outlined in this book and using built-in APIs are keys to future-proof code that will not be deprecated when a newer version of WordPress is released. We believe that every code snippet in this book will still be accurate and up-to-date for several years, just as several plugins we coded many years ago are still completely functional today.

# HOW THIS BOOK IS STRUCTURED

This book is, to date, one of the most powerful and comprehensive resources you can find about WordPress plugins. Advanced areas of the many WordPress APIs are covered, such as the REST API, cron jobs, and custom post types. This book is divided into three major parts. Reading the first five chapters is required if you are taking your first steps in the wonders of WordPress plugins. [Chapters 6](#) through [9](#) will cover most common topics in coding plugins, and understanding them will be useful when reading subsequent chapters. The remaining chapters cover advanced APIs and functions, can be read in any order, and will sometimes refer to other chapters for details on a particular function.

## CONVENTIONS

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

**WARNING** *Boxes with a warning label like this one hold important, not-to-be-forgotten information that is directly relevant to the surrounding text.*

**NOTE** *The note label indicates notes, tips, hints, tricks, and asides to the current discussion.*

As for styles in the text:

- We *italicize* new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.

- We show filenames, URLs, and code within the text like so: `persistence.properties`.

We present code in two different ways:

We use a monofont type with no highlighting for most code examples.

We use **bold** to emphasize code that is particularly important in the present context or to show changes from a previous code snippet.

## SOURCE CODE

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All the source code used in this book is available for download at [www.wiley.com/go/prowordpressdev2e](http://www.wiley.com/go/prowordpressdev2e) on the Downloads tab.

**NOTE** Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-1-119-66694-3.

## ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, such as a spelling mistake or faulty piece of code, we would be grateful for your feedback. By sending in errata, you may save another reader hours of frustration, and at the same time, you will be helping us provide even higher-quality information.

To find the errata page for this book, go to [www.wiley.com](http://www.wiley.com) and locate the title using the Search box. Then, on the book

details page, click the Errata link. On this page, you can view all errata that have been submitted for this book and posted by editors. If you don't spot “your” error on the Errata page, go to [support.wiley.com](http://support.wiley.com) and follow the directions to contact technical support and open a ticket to submit the error. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent printings of the book.

# 1

## An Introduction to Plugins

### WHAT'S IN THIS CHAPTER?

- Understanding what a plugin is
- Using available WordPress APIs
- Finding examples of popular plugins
- Separating plugin and theme functionality
- Managing and installing plugins
- Understanding types of WordPress plugins

WordPress is the most popular open source content management system available today. One of the primary reasons WordPress is so popular is the ease with which you can customize and extend WordPress through plugins. WordPress has an amazing framework in place that gives plugin developers the tools needed to extend WordPress in any way imaginable.

Understanding how plugins work, and the tools available in WordPress, is critical knowledge when developing professional WordPress plugins.

### WHAT IS A PLUGIN?

A *plugin* in WordPress is a PHP-based script that extends or alters the core functionality of WordPress. Quite simply, plugins are files installed in WordPress to add a feature, or set of features, to WordPress. Plugins can range in complexity from a simple social networking plugin to an extremely elaborate eCommerce package. There is no limit

to what a plugin can do in WordPress; because of this, there is no shortage of plugins available for download.

## How Plugins Interact with WordPress

WordPress features many different APIs for use in your plugin. Each API, or application programming interface, helps interact with WordPress in a different way. The following are the main available APIs in WordPress and their function:

- **Plugin:** Provides a set of hooks that enable plugins access to specific parts of WordPress. WordPress contains two different types of hooks: Actions and Filters. The Action hook enables you to trigger custom plugin code at specific points during execution. For example, you can trigger a custom function to run after a user registers a user account in WordPress. The Filter hook modifies text before adding it to or after retrieving it from the database.
- **Widgets:** Allows you to create and manage widgets in your plugin. Widgets appear under the Appearance ⇄ Widgets screen and are available to add to any registered sidebar in your theme. The API enables multiple instances of the same widget to be used throughout your sidebars.
- **Shortcode:** Adds shortcode support to your plugin. A shortcode is a simple hook that enables you to call a PHP function by adding something such as [shortcode] to a post or page.
- **HTTP:** Sends HTTP requests from your plugin. This API retrieves content from an external URL or for submitting content to a URL. Currently you have five different ways to send an HTTP request. This API standardizes that process and tests each method prior



to executing. Based on your server configuration, the API will use the appropriate method and make the request.

- **REST API:** Allows developers to interact with your WordPress website remotely by sending and receiving JavaScript Object Notation (JSON) objects. You can create, read, update, and delete (CRUD) content within WordPress. The REST API is covered extensively in [Chapter 12](#), “REST API.”
- **Settings:** Inserts settings or a settings section for your plugin. The primary advantage to using the Settings API is security. All settings data is scrubbed, so you do not need to worry about cross-site request forgery (CSRF) and cross-site scripting (XSS) attacks when saving plugin settings.
- **Options:** Stores and retrieves options in your plugin. This API features the capability to create new options, update existing options, delete options, and retrieve any option already defined.
- **Dashboard Widgets:** Creates Dashboard widgets. Widgets automatically appear on the WordPress Dashboard and contain all standard customization features including minimize, drag/drop, and screen options for hiding.
- **Rewrite:** Creates custom rewrite rules in your plugin. This API enables you to add static endpoints ( /custom-page/), structure tags ( %postname%), and feed links ( /feed/json/).
- **Transients:** Creates temporary options (cached data) in your plugins. This API is similar to the Options API, but all options are saved with an expiration time.
- **Database:** Accesses the WordPress database. This includes creating, updating, deleting, and retrieving

database records for use in your plugins.

- **Theme Customization (Customize) API:** Adds custom website and theme options to the WordPress Customizer. Theme customizations are displayed in a real-time preview prior to publishing to the live website.

There are additional, lesser known APIs that exist within the WordPress Core software. To view a full list, visit the Core Developer Handbook:

<https://make.wordpress.org/core/handbook/best-practices/core-apis>

WordPress also features pluggable functions. These functions enable you to override specific core functions in a plugin. For example, the `wp_mail()` function is a pluggable function. You can easily define this function in your plugin and send email using the Simple Mail Transfer Protocol (SMTP) rather than the default method. All pluggable functions are defined in the `/wp-includes/pluggable.php` WordPress Core file.

As an example, let's look at the `wp_mail()` pluggable function, which starts with this line of code:

```
if ( ! function_exists( 'wp_mail' ) ) :
```

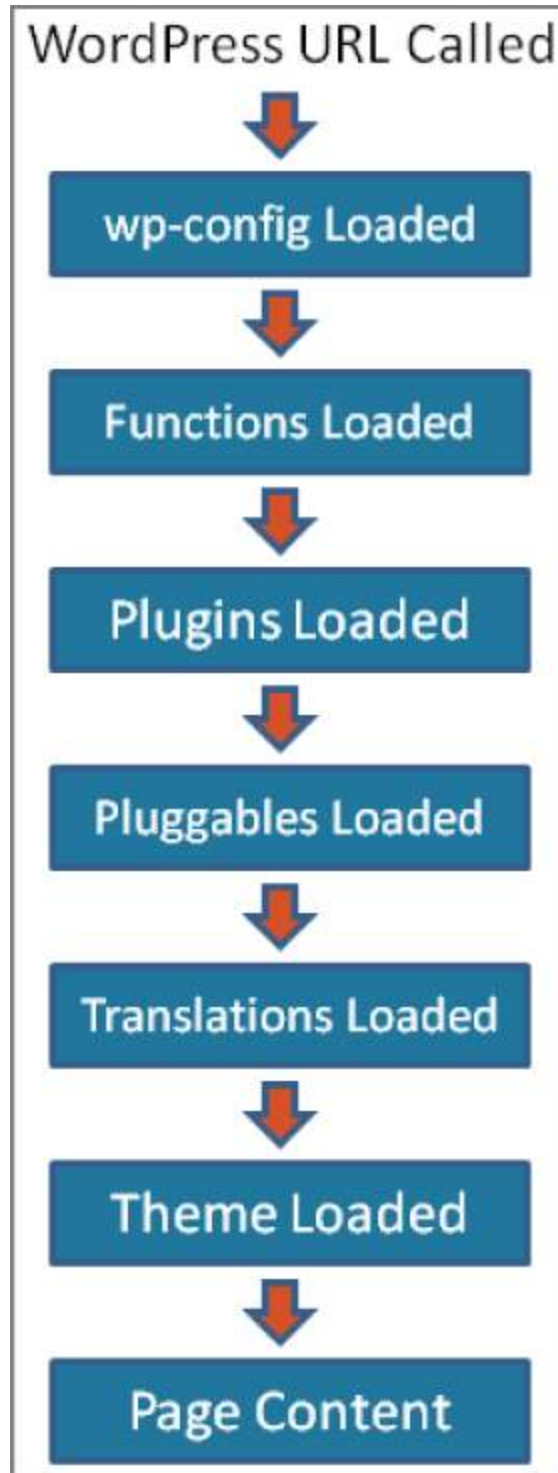
You can see that the code first checks to see whether a `wp_mail()` function already exists using the `function_exists()` PHP function. If you created your own custom `wp_mail()` function, that will be used; if not, the WordPress Core version of `wp_mail()` will be used.

**WARNING** *Pluggable functions are no longer being added to WordPress Core. Newer functions utilize hooks for overriding their functionality.*

You can use some predefined functions during specific plugin tasks, such as when a plugin is activated or deactivated and even when a plugin is uninstalled. [Chapter 2](#), “Plugin Framework,” covers these functions in detail.

## **When Are Plugins Loaded?**

Plugins are loaded early in the process when a WordPress-powered web page is called. [Figure 1-1](#) shows a high-level diagram of the standard loading process when loading a page in WordPress.



**FIGURE 1-1:** Loading a page in WordPress

The flow changes slightly when loading an admin page. The differences are minor and primarily concern what theme is loaded: admin theme versus your website theme.

# AVAILABLE PLUGINS

When researching available plugins, you need to know where to find WordPress plugins. You can download plugins from many places on the Internet, but this isn't always a good idea.

**WARNING** *As with any software, downloading plugins from an untrusted source could lead to malware-injected and compromised plugin files. It's best to download plugins only from trusted websites and official sources such as the official Plugin Directory.*

## Official Plugin Directory

The first place to start when researching available WordPress plugins is the official Plugin Directory at [WordPress.org](https://wordpress.org). The Plugin Directory is located at <https://wordpress.org/plugins>. With more than 55,000 plugins available and millions of plugin downloads, it's easy to see the vital role plugins play in every WordPress website. All plugins available in the Plugin Directory are 100 percent GPL and free to use for personal or commercial use.

## Popular Plugin Examples

Take a look at some of the more popular WordPress plugins available to get a sense of their diversity:

- **Yoast SEO:** Advanced search engine optimization functionality for WordPress. Features include custom metadata for all content, canonical URLs, custom post type support, XML sitemaps, and more!
  - <https://wordpress.org/plugins/yoast-seo>

- **WPForms:** A powerful drag-and-drop form builder. Create simple contact forms and powerful subscription payment forms, all without writing a single line of code.
  - <https://wordpress.org/plugins/wpforms-lite>
- **BuddyPress:** A suite of components used to bring common social networking features to your website. Features for online communities include member profiles, activity streams, user groups, messaging, and more!
  - <https://wordpress.org/plugins/buddypress>
- **WooCommerce:** Advanced eCommerce solution built on WordPress. This is an extremely powerful plugin allowing anyone to sell physical and digital goods online.
  - <https://wordpress.org/plugins/woocommerce>
- **Custom Post Type UI:** Easy-to-use interface for registering and managing custom post types and taxonomies in WordPress.
  - <https://wordpress.org/plugins/custom-post-type-ui>

As you can see, the preceding plugins can handle a variety of complex tasks. The features added by these plugins are universal and features that many websites on the Internet could have.

## Popular Plugin Tags

Now you will look at some popular tags for plugins. Plugin tags are just like blog post tags, simple keywords that describe a plugin in the Plugin Directory. This makes it easy to search for existing plugins by tag. The following are popular examples: