



Complete Step-by-Step Guide for Beginners

■ Audience	■ Chapters	■ ■ Language	■ Format
Absolute Beginners	8 Complete Steps	PHP + WordPress API	Hands-On Learning

■ What You Will Learn in This Guide

- ✓ Understand what a WordPress Plugin is and how it works
- ✓ Set up your development environment (tools & folders)
- ✓ Write your very first plugin with PHP code
- ✓ Use WordPress Hooks — Actions & Filters
- ✓ Create Admin Settings Page for your plugin
- ✓ Save & retrieve data using WordPress Database
- ✓ Package your plugin and upload it to WordPress
- ✓ Test, Debug and Publish your plugin

■ Table of Contents

STEP 1	What is a WordPress Plugin?	<i>Understanding the basics</i>	Pg 3
STEP 2	Setting Up Your Environment	<i>Tools, software & folder structure</i>	Pg 4
STEP 3	Creating Your First Plugin	<i>Plugin header & basic PHP file</i>	Pg 5
STEP 4	WordPress Hooks — Actions & Filters	<i>The core of plugin development</i>	Pg 7
STEP 5	Building an Admin Settings Page	<i>Dashboard menu & options</i>	Pg 9
STEP 6	Saving Data to Database	<i>Options API & custom tables</i>	Pg 11
STEP 7	Packaging & Uploading Your Plugin	<i>ZIP file & WP admin upload</i>	Pg 13
STEP 8	Testing, Debugging & Publishing	<i>Best practices & WordPress.org</i>	Pg 14

1

What is a WordPress Plugin?

Understanding the foundation before we code

Simple Definition

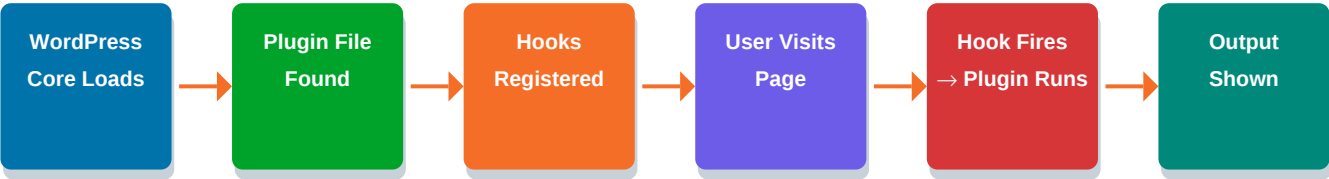
A **WordPress Plugin** is a small software package that you can add to your WordPress website to give it new features or change how it works — *without touching the core WordPress files*. Think of it like installing an app on your phone to add new functions.

■ Real-Life Analogy

WordPress = Your Smartphone
Plugin = An App you install

Just like WhatsApp adds messaging to your phone, a plugin adds features to your WordPress site.

How Plugins Work — The Big Picture



Popular Plugin Examples

Plugin Name	What It Does	Downloads
WooCommerce	Adds online store / shopping cart	5 Billion+
Yoast SEO	Helps your site rank on Google	3 Billion+
Contact Form 7	Creates contact forms easily	2 Billion+
WP Super Cache	Makes your website load faster	1 Billion+
Akismet	Blocks spam comments automatically	1 Billion+

2

Setting Up Your Environment

Install the right tools before writing any code

Tools You Need (All Free!)

■ ■ Tool	■ Purpose	■ Download
XAMPP / MAMP	Local server to run WordPress on your computer	apachefriends.org
VS Code	Best free code editor for writing PHP	code.visualstudio.com
WordPress (latest)	Download fresh copy of WordPress	wordpress.org/download
FileZilla (optional)	Upload files via FTP to live server	filezilla-project.org
Browser DevTools	Debug and test in Chrome/Firefox	Built-in (Press F12)

Plugin Folder Structure

Every plugin lives inside the **wp-content/plugins/** folder. Below is the correct folder structure for your plugin:

```
wp-content/  
  ■ ■ plugins/  
    ■ ■ my-first-plugin/  
      ■ ■ my-first-plugin.php  
      ■ ■ readme.txt  
      ■ ■ assets/  
        ■ ■ css/  
          ■ ■ style.css  
        ■ ■ js/  
          ■ ■ script.js  
        ■ ■ images/  
      ■ ■ includes/  
        ■ ■ class-plugin-name.php  
        ■ ■ functions.php  
      ■ ■ admin/  
        ■ ■ admin-page.php
```

■ Quick Setup Checklist

- Install XAMPP/MAMP and start Apache + MySQL
- Download WordPress and extract to htdocs folder
- Create new database in phpMyAdmin
- Run WordPress installer at <http://localhost/yoursite>
- Go to wp-content/plugins/ and create your plugin folder
- Open the folder in VS Code – you are ready!

3

Creating Your First Plugin

Write the plugin header and basic code

The Plugin Header — Most Important Part!

Every WordPress plugin MUST start with a special comment block called the **Plugin Header**. WordPress reads this to identify your plugin. Without it, WordPress will NOT recognize your file as a plugin.

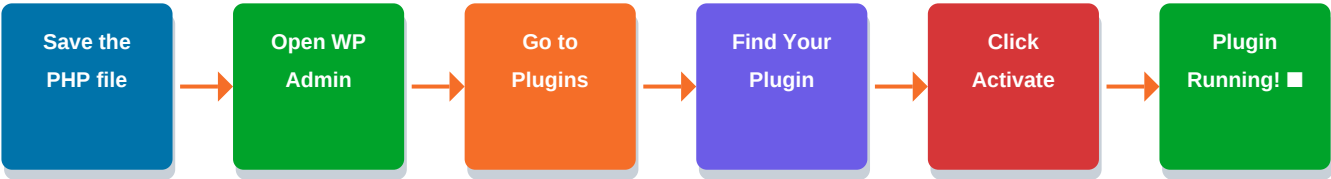
■ my-first-plugin.php — Plugin Header (Required)

```
<?php
/**
 * Plugin Name: My First Plugin
```

Understanding Each Header Line

Header Field	What It Means	Required?
Plugin Name	Display name shown in WP Admin plugins list	YES ✓
Plugin URI	Link to your plugin's homepage	No
Description	Short description (shown in plugins list)	Recommended
Version	Version number (use 1.0.0 format)	Recommended
Author	Your name or company name	Recommended
Author URI	Link to your personal/company website	No
License	How others can use your code (GPL-2.0+ is standard)	Recommended
Text Domain	Unique ID for translations (use plugin folder name)	Recommended

Activate Your Plugin



4

WordPress Hooks — Actions & Filters

The most powerful concept in plugin development

What are Hooks?

Hooks are the **backbone of WordPress plugin development**. They allow your plugin to *hook into* WordPress at specific moments and run your custom code. There are 2 types of hooks:

■ ACTION Hooks

Do something at a specific point. No return value needed.

```
add_action( $hook, $function );
```

Examples: • wp_head • init • admin_menu • save_post • wp_enqueue_scripts

■ FILTER Hooks

Modify a value and return it. Always returns modified data.

```
add_filter( $hook, $function );
```

Examples: • the_content • the_title • excerpt_length • login_message • wp_title

Action Hook — Code Example

■ Adding a notice to the footer using an Action Hook

```
<?php
// Step 1: Create your custom function
function mfp_add_footer_notice() {
```

Filter Hook — Code Example

■ Adding text to every post using a Filter Hook

```
<?php
// Step 1: Create a function that receives AND returns content
function mfp_add_post_suffix( $content ) {
```

5

Building an Admin Settings Page

Add your plugin to the WordPress Dashboard menu

Why Add a Settings Page?

A Settings Page lets your plugin users **change options without touching code**. For example: changing text, colors, enabling/disabling features — all through a nice form in the WP Admin dashboard.

Step-by-Step: Add Admin Menu + Settings Page

■ Complete Admin Settings Page Code

```
<?php
/* ■■■ STEP A: Register the Admin Menu ■■■ */
function mfp_add_admin_menu() {
```

■■ STEP C: Register Settings & Add a Text Field

```
function mfp_register_settings() {
    // 1. Register the setting (saves to wp_options table)
    register_setting( 'mfp_settings_group', 'mfp_custom_text' );
```


7

Packaging & Uploading Your Plugin

Create a ZIP file and install it on any WordPress site

Method 1 — Upload via WordPress Admin (Easiest)

1

Create the ZIP File

Right-click your plugin folder (e.g. my-first-plugin/) → Send to → Compressed (zip) folder. Make sure the folder is inside the ZIP, not just the files!

2

Go to WordPress Admin

Open your WordPress dashboard at: yoursite.com/wp-admin Login with your admin username and password.

3

Navigate to Plugins

In the left sidebar: click Plugins → Add New Plugin

4

Upload the ZIP

Click the 'Upload Plugin' button at the top → Choose File → select your .zip file → click 'Install Now'

5

Activate Plugin

After installation, click 'Activate Plugin'. Your plugin is now live on the site!

Method 2 — Manual Upload via FTP

■ FTP Upload Steps (for when Admin upload doesn't work)

1. Open FileZilla and connect to your server (host, username, password from hosting)
2. On the RIGHT panel, navigate to: `/public_html/wp-content/plugins/`
3. On the LEFT panel, find your plugin folder on your computer
4. Drag the FOLDER (not individual files) to the plugins directory
5. Go to WP Admin → Plugins → Find your plugin → Click Activate

■■ Common Upload Mistakes to Avoid

- x Don't zip individual files – always zip the FOLDER
- x Don't use spaces in folder or file names (use hyphens: my-plugin)
- x Don't upload a folder inside another folder (double nesting)
- x Make sure your main PHP file has the Plugin Header comment
- x Plugin won't show? Check that the file name matches the folder name

8

Testing, Debugging & Publishing

Polish your plugin and share it with the world

Enable Debug Mode in WordPress

■ Add these lines to wp-config.php for debugging

```
// Open wp-config.php in your root WordPress folder  
  
define( 'WP_DEBUG',          true ); // Enable debug mode
```

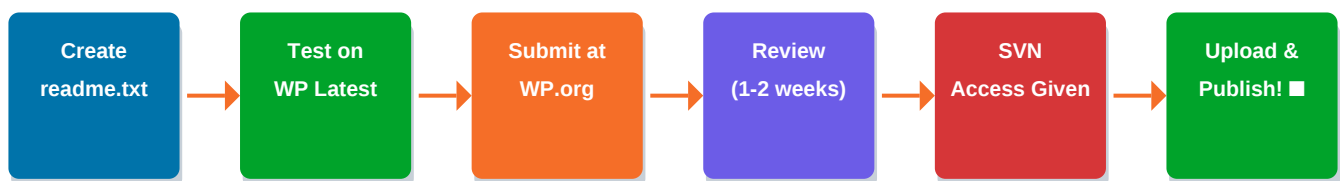
Testing Checklist

- ✓ Activate plugin — no PHP errors or blank white screen?
- ✓ Deactivate and reactivate — still works correctly?
- ✓ Settings page saves and loads data properly?
- ✓ Plugin works with default WordPress theme (Twenty Twenty-Four)?
- ✓ Test on a fresh WordPress install — no other plugins running
- ✓ Check debug.log for any PHP warnings or notices
- ✓ Test all form inputs — try empty values and special characters
- ✓ Uninstall plugin — does it clean up its data from the database?

Security Best Practices (Very Important!)

Security Rule	Code Example	Why?
Nonce Verification	<code>check_admin_referer('my_nonce')</code>	Prevent CSRF attacks
Sanitize Input	<code>sanitize_text_field(\$_POST['field'])</code>	Remove harmful data
Escape Output	<code>esc_html(\$variable)</code>	Prevent XSS attacks
Check Permissions	<code>current_user_can('manage_options')</code>	Only admins can act
Block Direct Access	<code>if(!defined('ABSPATH')) exit;</code>	Stop direct PHP access

Publish on WordPress.org (Free Plugin Directory)



■ readme.txt — Required for WordPress.org Submission

=== My First Plugin ===

Contributors: yourwpusername

Tags: tool, utility, simple

Requires at least: 5.0

Tested up to: 6.5

Requires PHP: 7.4

Stable tag: 1.0.0

License: GPLv2 or later

== Description ==

A short description of what your plugin does.

== Installation ==

1. Upload to `/wp-content/plugins/` directory
2. Activate the plugin through the Plugins menu

== Changelog ==

= 1.0.0 = Initial release.

■ Quick Reference Cheat Sheet

FUNCTION	WHAT IT DOES	EXAMPLE USE CASE
<code>add_action()</code>	Run code at a specific WP event	Add text to footer
<code>add_filter()</code>	Modify and return a WP value	Change post content
<code>register_activation_hook()</code>	Run code when plugin activates	Create DB table
<code>register_deactivation_hook()</code>	Run code when plugin deactivates	Clear cache
<code>register_uninstall_hook()</code>	Clean up when plugin is deleted	Delete options
<code>get_option()</code>	Retrieve a saved setting	Get user preference
<code>update_option()</code>	Save a setting to database	Store form data
<code>delete_option()</code>	Remove a saved setting	Cleanup on uninstall
<code>add_menu_page()</code>	Add top-level admin menu	Plugin settings page
<code>add_submenu_page()</code>	Add sub-item to admin menu	Plugin sub-section
<code>wp_enqueue_script()</code>	Add JavaScript to pages	Load your plugin.js
<code>wp_enqueue_style()</code>	Add CSS to pages	Load your style.css
<code>current_user_can()</code>	Check user permissions	Restrict admin actions
<code>sanitize_text_field()</code>	Clean/sanitize user input	Before saving to DB
<code>esc_html()</code>	Make output safe to display	When echoing variables
<code>\$wpdb->insert()</code>	Insert row into DB table	Save user submission
<code>\$wpdb->get_results()</code>	Query DB and get results	Show data from table

■ Helpful Resources & Next Steps

■ Official WordPress Developer Docs	developer.wordpress.org/plugins
■ WordPress Hook Reference	developer.wordpress.org/reference/hooks
■ WordPress Plugin Directory	wordpress.org/plugins
■ WordPress Support Forums	wordpress.org/support
■ WP Beginner Tutorials	wpbeginner.com/category/wp-tutorials
■ Learn WordPress	learn.wordpress.org