

RACHEL McCOLLIN

WORDPRESS®

 PUSHING THE LIMITS

Fresh & Clean



WILEY

WordPress[®]: **Pushing the Limits**

Rachel McCollin

WILEY

This edition first published 2013

© 2013 John Wiley & Sons, Ltd.

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

DESIGNATIONS USED BY COMPANIES TO DISTINGUISH THEIR PRODUCTS ARE OFTEN CLAIMED AS TRADEMARKS. ALL BRAND NAMES AND PRODUCT NAMES USED IN THIS BOOK ARE TRADE NAMES, SERVICE MARKS, TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNERS. THE PUBLISHER IS NOT ASSOCIATED WITH ANY PRODUCT OR VENDOR MENTIONED IN THIS BOOK. THIS PUBLICATION IS DESIGNED TO PROVIDE ACCURATE AND AUTHORITATIVE INFORMATION IN REGARD TO THE SUBJECT MATTER COVERED. IT IS SOLD ON THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING PROFESSIONAL SERVICES. IF PROFESSIONAL ADVICE OR OTHER EXPERT ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL SHOULD BE SOUGHT.

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/ or its affiliates in the United States and/or other countries, and may not be used without written permission. WordPress is a registered trademark of WordPress Foundation. All other trademarks are the property of their respective owners. John Wiley & Sons, Ltd. is not associated with any product or vendor mentioned in the book.

A catalogue record for this book is available from the British Library.

ISBN 978-1-118-59719-4 (paperback); ISBN 978-1-118-59717-0 (ebook); 978-1-118-59718-7 (ebook); 978-1-118-59715-6 (ebook)

Set in 9.5 Myriad Pro by Wiley Indianapolis Composition Services

Printed in United States by Bind-Rite

Publisher's Acknowledgements

Some of the people who helped bring this book to market include the following:

Editorial and Production

VP Consumer and Technology Publishing Director: Michelle Leete

Associate Director–Book Content Management: Martin Tribe

Associate Publisher: Chris Webb

Associate Commissioning Editor: Ellie Scott

Project Editor: Tom Dinse

Copy Editor: Luann Rouff

Technical Editor: Mike Little

Editorial Manager: Jodi Jensen

Senior Project Editor: Sara Shlaer

Editorial Assistant: Annie Sullivan

Marketing

Associate Marketing Director: Louise Breinholt

Marketing Manager: Lorna Mein

Senior Marketing Executive: Kate Parrett

Marketing Assistant: Tash Lee

Composition Services

Compositor: Jennifer Goldsmith

Proofreader: Wordsmith Editorial

Indexer: Potomac Indexing, LLC

About the Author

Rachel McCollin is a keen (if sometimes slow) cyclist, amateur photographer, and mother of two budding boy geeks.

She first discovered WordPress in 2010 when a client asked for a CMS to manage their site, and hasn't looked back since. She attended her first WordCamp a few months later and is now a regular fixture at WordCamp UK, having been declared "the best speaker wearing red shoes" in 2012.

She contributes to *Smashing Magazine* and [wptutsplus](#) and is the author of two WordPress-related Beginners Guides for Packt Publishing. She also runs Compass Design, a WordPress web development agency in Birmingham, UK.

About the Technical Editor

Mike Little is the co-founder of WordPress and founder and CEO of zed1.com ltd, a WordPress specialist web development consultancy.

He is a 100% WordPress specialist providing WordPress development, consultancy, training, and tech editing services.

He has developed many web sites for his clients including a number of UK government web sites and the award-winning I'm a Scientist, Get Me Out of Here (<http://imascientist.org.uk>) educational science engagement site. All in WordPress, of course. He is a long-term software developer (25+ years), with many programming languages, many disciplines, and many years under his belt, most at senior levels. He is a published author and technical editor.

He runs his local Manchester WordPress User Group, and serves on the WordCamp UK organising committee.

He lives with his family in Stockport, UK.

Acknowledgements

I'd like to thank all those in the WordPress community who have helped me develop my own use of WordPress, in particular the organizers of WordCamp UK, without whom I wouldn't have had the opportunity to meet many of the developers I'm now pleased to call friends.

I'd also like to thank Birgit Gruber (formerly of Wiley) for bringing me on board with this project, and Chris Webb, Tom Dinse, Luann Rouff, Sara Schlaer, and Ellie Scott for their professionalism and support. A huge thanks goes to Mike Little for his enthusiasm and feedback—having a co-founder of WordPress involved with this book is a pleasure and an honor. Members of my team at Compass Design provided feedback on the initial ideas for this book, inspired some of the content and contributed code used in the case studies—thanks to Mark Wilkinson, Tammie Lister, Scott Evans, and Nivi Morales.

My husband, Pete, spent many hours keeping two small boys calm while I disappeared to quiet rooms, libraries, or coffee shops to write, and I thank him for his continuing and unwavering support and patience.

Finally, I'd like to thank Jonny Allbut, whose input in planning this book made it much better than it would have been without him—I hope we manage to work on another writing project together sometime.

Contents

Introduction	1
Part I: Professional WordPress Development	5
Chapter 1 WordPress As a Professional Web Development Tool	7
What It Means to Be a Professional WordPress Developer	7
Professional Coding Practices	8
Approaching Large-Scale Site Design	13
Developing Your WordPress Skills: It Never Stops	17
Giving Something Back to the WordPress Community	18
Testing	19
Writing Free Plugins and Themes	19
Contributing to the WordPress Website	19
Getting Involved in WordPress Groups or WordCamps	20
The Potential for Professional WordPress Development	20
Harnessing WordPress As a CMS	21
WordPress for SEO and Site Conversions	22
Developing Themes and Plugins You Can Sell	22
Summary	23
Further Resources	24
Professional WordPress Development	24
Coding Practices	24
Giving Something Back to the WordPress Community	25
Developing Your WordPress Skills	25
WordPress for SEO	26
Developing Themes and Plugins You Can Sell	26
Chapter 2 Kicking Off a WordPress Project	27
Setting Up Development Hosting and Domains	27
Selecting the Best Hosting and Domain Setup for Your Project	27
Agreeing What to Put in Place During Development	28
SEO During Site Development	29

Content Planning and Structure	36
Identifying How to Structure Content: Questions for Clients.	36
Planning Content Generation.	38
Matching Content to the Best WordPress Page or Post Type	39
Planning Content Display.....	40
Writing a Technical Brief	40
Configuring WordPress for the Team and the Client	43
Planning Access and User Capabilities	43
Summary.....	45
Further Resources	45
SEO and Teaser Pages	45
Project Collaboration: Themes and Plugins.....	45
Site Development Plugins	46
Part II Content and Administration	47
Chapter 3 Content Organization and Relationships	49
Rationalizing Your Content	50
Considering a Site's Layout and Structure.....	50
Identifying Editing Requirements	53
Displaying Device-Specific Content.....	54
Examining How WordPress Stores Content	56
The Core Database Structure	57
The WordPress Options Table.....	59
Building Custom Post Types	59
Manipulating Post Type Parameters.....	62
Building Relationships Between Content.....	66
Adding Extra Data to Content	67
Working with Taxonomies.....	68
Custom Fields and Post Metadata	73
Adding Media to Content	81
How Post Attachments Work.....	81
Featured Images	87
Summary	88
Further Resources	89
Device-Specific Content	89
How WordPress Stores Content.....	89
Custom Post Types	89
Taxonomies	89
Custom Fields and Post Metadata	90

Chapter 4 Customizing the WordPress Admin.....	91
When to Customize the WordPress Admin	91
Client Sites	91
Themes and Plugins	92
WordPress Roles and Capabilities	92
Custom Roles and Capabilities	97
Creating Admin Screens	105
Customizing the Dashboard	106
Creating Admin Sections and Screens	113
The Theme Customizer	125
Content and Editing	130
Customizing Listing Screens	130
Providing Help for Editors.....	132
Sanitizing Data	134
Callback Functions to Sanitize Data	134
Summary.....	136
Further Resources	136
Roles and Capabilities	136
Admin Screens and the Settings API	137
The Theme Customizer	137
Listings Screens	138
Adding Meta Boxes.....	138
Data Sanitization	138
Part III Practicalities of Developing and Hosting WordPress Sites.....	139
Chapter 5 Development and Hosting Environments	141
Creating a Robust Development Environment	141
The Advantages of Local Development	142
Working Collaboratively on Large Projects	144
Virtual Machines and Browser Testing.....	148
Testing on Other Devices	149
Development Software and Tools	150
Creating a Client Approval Site	151
The Approval Process.....	151
Controlling Access and Search Engine Visibility	152
Client Approval Access and Privileges.....	153
Tracking and Responding to Amendments	154

Migrating a Site for Launch	154
Preparing to Go Live	154
Going Live Site Checklist	155
Moving WordPress Sites	156
Hosting Options for Clients	158
What You Need to Know	158
Finding and Getting the Most from a Hosting Provider	160
Summary	161
Further Resources	161
Setting Up a Local Development Environment	161
Tools for Collaborative Working	162
Browser Testing Tools	163
Browser Support Tools	163
Mobile Testing Tools	163
Moodboard and Color Palette Tools	163
Bug Tracking Tools	164
Moving a WordPress Site	164
Hosting	164
Chapter 6 Avoiding and Dealing with Disaster	165
What Could Go Wrong and the Consequences	165
Avoiding Problems	166
Site Performance Problems	166
Avoiding Data Loss	169
Updating WordPress and Installed Plugins Safely	172
Improving Security	174
Planning for the Future	177
WordPress Upgrades Overview	177
Using the WordPress Trac Site	177
Testing with Pre-release WordPress Versions	178
Testing and Debugging Plugins and Themes	180
Detecting and Supporting Multiple Versions of WordPress	180
When Disaster Strikes	182
Communicating with Clients or Users	182
Identifying the Problem	183
Fixing the Problem	185
Summary	189
Further Resources	189
Server Monitoring	189
Backing Up	189

WordPress Upgrades and Planning for the Future.....	189
WordPress Security	190
Dealing with Disaster	190

Part IV Pushing the Limits: The Best Tools for Site Development.....191

Chapter 7 Theme Building: Frameworks, Standalone Themes, or Child Themes?	193
Theme-Building Principles.	193
Approaches to Theme Building.....	194
Rapid Theme Development for Teams.....	195
Building Themes from Scratch.	196
Making It Work	197
Customizing an Existing Theme	198
Making It Work	199
Using Child Themes.	200
The Parent/Child Theme Relationship.....	200
Choosing Your Parent Theme.....	203
Developing with Theme Frameworks.	204
So Many Frameworks!	205
Getting More from Frameworks.....	212
Building Your Own Theme Framework.....	219
Theme Building for Networks.	231
Differences Between Single and Multisite.....	231
Planning How to Use Network Themes	231
BuddyPress and Beyond.	232
BuddyPress and Themes.....	233
Summary	235
Further Resources.	236
Theme Building	236
Adapting an Existing Theme.....	236
Parent and Child Themes.....	236
Theme Frameworks	236
Framework Functions	237
Framework Action Hooks.....	237
Framework Filter Hooks.....	237
Building Your Own Framework	238
Action Hooks.....	238
Filter Hooks.....	238

Widgets	238
Menus	239
WordPress Multisite	239
BuddyPress	239
Chapter 8 Conditional Display: Functions, Template Files, and Styling	241
Conditional Content: When It Applies	241
Using Conditional Tags	242
Theme Files for Conditional Content	249
Conditional Styling	257
Conditional Content in Action: Case Study	263
Summary	270
Further Resources	271
Conditional Tags	271
Template Files and Includes	271
Conditional Styling	272
Chapter 9 Custom Functionality in Theme Functions and Plugins	273
Theme Functions vs. Plugins	273
Working with the Theme Functions File	275
Uses for the Functions File	275
Coding a Robust Functions File	275
Adding Include Files for Functions	276
Functions Files and Child Themes	277
Building Functionality into a Plugin	282
Plugins Overview	282
Coding Robust Plugins	286
Plugin Security	289
Hooking Your Plugin to WordPress	293
Creating Settings, Parameters, and Controls	295
Summary	309
Further Resources	309
Plugins vs. Themes	309
Theme Functions File	309
Plugins Overview	310
WordPress APIs	310
Plugin Security	311
Building Widgets	311
Creating Shortcodes	311
Plugin Styling	312

Chapter 10 Making Your Site Soar: HTML, JavaScript, and CSS	313
Choosing Your Approach	313
Optimizing Your Code	314
HTML	314
Optimizing Stylesheets and Scripts	316
Working with JQuery	319
Practical Applications	320
Creating Forms	320
Media and Galleries	328
Approaches to Animations	339
Improving Navigation	343
Summary	347
Further Resources	347
Optimizing Performance	347
HTML5	347
Web Standards	348
Accessibility	348
Stylesheets and Scripts	348
Forms	348
Media and Galleries	349
CSS Animations	349
JQuery Animations	350
Navigation	350
Chapter 11 Device Compatibility: Responsive and Mobile Development	351
WordPress As a Responsive Platform: Trends and Advances	352
WordPress Makes Responsive Design Easier	353
Identifying the Best Approach for Your Clients	353
Developing a Responsive Theme Using CSS	356
Fluid Layout and Media Queries for Layout	356
Mobile-First Themes	365
Child Themes and Responsive Design	366
Responsive Navigation	369
Conditional Content for Devices	375
When to Send Different Content to Different Devices	375
Techniques for Identifying Device Types	376
Using Responsive Images	378
Conditional Navigation	381
Mobile-Only Themes	383
Uses for a Mobile Theme	383
Displaying a Different Theme Based on Device	384

Building Your Mobile Theme	384
Exploring the Possibilities: WordPress Web Apps	385
Summary	397
Further Resources	397
Responsive Design	397
Conditional Content for Mobile	398
Responsive Images	398
WordPress Web Apps	398
Chapter 12 Releasing Your Code to the Public	399
Sharing Your Code	399
Filling a Gap in the Market	400
Free vs. Premium Themes and Plugins	401
Releasing Your Code on wordpress.org	401
The Theme Review Guidelines	401
Plugin Submission Guidelines	403
Preparing Your Code for Release	404
Maintenance and Support	413
Supporting Your Theme or Plugin	413
Updates and Maintenance	414
Selling Premium Themes and Plugins	415
Commercial Development Expectations	416
The Practicalities	416
Preparing for Translation	419
Why Translate Your Code?	419
Internationalization and Localization	420
WordPress Translation Functions	422
Summary	426
Further Resources	426
Theme and Plugin Guidelines	426
Theme Unit Testing	426
Preparing Your Code for Release	427
Documentation	427
Support and Maintenance	427
Premium Themes and Plugins	428
Translation	428
Index	429

Introduction

WordPress is the world’s most popular content management system (CMS), and it continues to grow in popularity. There’s an increasing trend for owners of websites of all descriptions to move to WordPress as their needs scale, along with their frustration with other customized or unintuitive CMSs that they may be using. As Matt Mullenweg said in 2012:

“We might not always be the platform people start with, but we want to be what the best graduate to.”

WordPress is now far more than a blogging platform. Enhancements such as custom post types and taxonomies, as well as improvements to the Dashboard such as the Menus screen added in version 3.0, and the new Media Manager enhancements in version 3.5, mean that developers can push WordPress further and users can quickly create beautiful, customized websites. If you’re prepared to write custom functions and hook into the WordPress APIs, then there’s very little you can’t do.

WordPress is free and open source, with a large and dedicated community of users and developers, which is one of its strengths. It’s also a platform on which you can build a career and a business. As a WordPress developer you may be considering setting up your own agency building WordPress-powered sites, or you may have already done this and want to take your skills to the next level so you can meet the needs of larger clients with more complex requirements.

This book will help you advance your WordPress skills so you can become a sought-after WordPress professional—working with teams and clients to create unique and powerful projects.

Who This Book Is For

This book is for WordPress developers. It won’t teach you how to set up a WordPress installation and add themes, plugins, and content to create a basic site. You can find many other books out there that do that.

The book is aimed at people who have experience with web development and developing with WordPress. You might be an experienced plugin or theme developer, or you may have built sites for a handful of clients. You could be working as part of an agency team using WordPress, or in the web department of an organization whose own site is powered by WordPress. Or you could simply be an enthusiastic WordPress developer who wants to take things further.

Most of the examples in this book will teach you how to code custom PHP to make WordPress do what you want—you won’t learn how to use the WordPress admin screens here. Although some screenshots of the admin are included, you won’t learn which button to click or which menu to select—it is assumed that you already know this.

To get the most from this book, then, you should have a solid understanding of HTML, some CSS experience, and be capable of writing your own PHP. All the PHP in the examples is explained in detail but the book does not provide an introduction to PHP (again, there are other books that do that).

What This Book Covers

As WordPress becomes more popular and more widely accepted as a serious CMS, there is an increasing need for skilled developers who can use it to build custom websites that are much more than a blog or simple “brochureware.”

In addition, as large organizations switch over to WordPress, there is a need for WordPress developers who can manage large-scale web development projects, understand client needs, lead teams of developers, and develop robust, future-proof sites.

This book focuses on both of these. In the first part you will learn about WordPress as a professional tool, and about the skills you need to manage WordPress projects. You won’t get a complete primer on business or management, but you will be prompted to think about how your WordPress skills need to be complemented by project management, planning, and team management skills. In the subsequent parts you will learn techniques for pushing WordPress to create powerful, customized sites. You will learn how to customize the admin and enhance security, as well as how to build mobile or responsive sites and make the best use of JavaScript and CSS alongside PHP. You’ll learn when to add functionality via themes and when to use plugins, and make use of conditional tags for styling and content. Finally, you’ll learn about releasing your code to the public, so you can test, submit, and market your themes and/or plugins and prepare them for translation.

How This Book Is Structured

This book isn’t designed to be read from cover to cover (although please feel free to do so!). You can dip into chapters relevant to your expertise and learning needs, as each chapter stands alone. However, the book is divided into four related parts.

Part I: Professional WordPress Development

If you’re an experienced WordPress developer but have little experience using WordPress professionally or managing WordPress projects, this part of the book will introduce you to the relevant considerations and skills.

- **Chapter 1: “WordPress As a Professional Web Development Tool”**—Why WordPress is great tool for professional web development and for building a career. Learn about the potential for professional WordPress development—it isn’t just about client work.
- **Chapter 2: “Kicking Off a WordPress Project”**—How to begin WordPress projects in a way that gives you the best chance of success, including planning, writing a technical brief, and setting up your development environment.

Part II: Content and Administration

This part explains how WordPress manages content and data, and it will provide the skills you need to customize the WordPress admin for your clients.

- **Chapter 3: “Content Organization and Relationships”**—An analysis of how WordPress uses tables to store different kinds of content and the relationships between these tables, and what it means for development.

- **Chapter 4: “Customizing the WordPress Admin for Client Sites”**— How to customize the admin screens to make things easier for your clients and give them access to settings and options. This chapter also covers the Settings API.

Part III: Practicalities of Developing and Hosting WordPress Sites

As covered in this part, ensuring that you have robust hosting and development environments and can protect your code from security attacks or other problems enhances your reputation as a developer and saves you a lot of time.

- **Chapter 5: “Development and Hosting Environments”**—How to identify the most appropriate development and hosting environments for your projects, set them up, and configure them.
- **Chapter 6: “Avoiding and Dealing with Disaster”**— Everyone dreads a site being hacked or breaking after a WordPress update. This chapter shows you how to anticipate potential problems and avoid them, as well as how to deal with disaster if it does strike.

Part IV: Pushing the Limits: The Best Tools for Site Development

One of WordPress’ many strengths is that you often have more than one way to accomplish a task. Using specific areas of development, this part of the book looks at the different options you have to achieve your objectives. Sometimes there is a “better” approach, but on other occasions it comes down to the specifics of the project or to your own, your client’s, or your team’s experience and preferences.

- **Chapter 7: “Theme Building: Frameworks, Standalone Themes, or Child Themes?”**—There are a few valid approaches to developing themes, and none of them is necessarily the “best” one. In this chapter you’ll learn about the different approaches and when each might be most useful; how to interact with the popular theme frameworks; and how to build your own framework using custom action hooks and filter hooks.
- **Chapter 8: “Conditional Content: Functions, Template Files, and Styling”**—Different sections within a site may need to work in very different ways, either functionally or visually. In this chapter you’ll learn how to use a variety of methods to achieve this.
- **Chapter 9: “Custom Functionality in Theme Functions and Plugins”**—Sometimes it’s appropriate to add functionality to your theme’s functions file, but in many cases writing a plugin will be better. In this chapter you’ll learn how to identify which approach to use and how to build plugins to add commonly required functionality.
- **Chapter 10: “Making a Site Soar: HTML, JavaScript, and CSS”**— WordPress sites don’t run on PHP alone. To build interactivity, styling, and media into your projects you’ll need to work with CSS and JavaScript, and no theme will function without HTML. In this chapter you’ll learn how to use these languages within WordPress and how to add animations, enhance the use of media, and customize navigation menus.
- **Chapter 11: “Device Compatibility: Responsive and Mobile Development”**— Responsive design has now entered the mainstream. In this chapter you’ll learn how to add a responsive layout to your themes, how to work mobile first, and when to use a separate mobile theme for custom mobile sites.
- **Chapter 12: “Releasing Your Code to the Public”**—There are many opportunities for developers to share their code, either free or for payment. This chapter explores requirements for themes and plugins in the WordPress repository and how to submit them, and looks at the practicalities of selling your code for money, including how to identify a need and how to market and distribute your code. Finally, you’ll learn how to prepare your code for translation.

What You Need to Use This Book

All examples in this book have been developed using WordPress version 3.5. In some cases the Twenty Twelve theme (<http://wordpress.org/extend/themes/twentytwelve>) has been used to power sites used in the examples.

In order to follow along, you'll need the following:

- A local or remote installation of WordPress 3.5 or later (preferably the latest version)
- A text editor (with FTP access if you're working remotely)
- A modern browser

You'll have your own preferred text editor and browser, so I won't make recommendations.

WordPress Documentation

The best source of documentation on WordPress is the codex at <http://codex.wordpress.org/>. Each chapter also contains a "Further Reading" section at the end, and in many cases the links quoted are to pages in the Codex. Codex pages are continually edited and added to, so if you can't find any of the resources listed, a search of the Codex is your best bet.

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 of the source code used in this book is available for download at www.wiley.com/go/pt1/wordpress. For example, you will find the following sample code online in the Chapter 10 folder, in the Example project, and the `example.php` file.

Example.php

```
<?php
function wppt1_call_scripts3() {
    wp_enqueue_script( 'example_script', plugins_url('/js/example-script.js',
    __FILE__), , true );
}
add_action('wp_enqueue_scripts', 'wppt1_call_scripts3' );
?>
```

Some source code snippets shown in the book are not comprehensive, as they are meant to help you understand a particular concept in the chapter. For these instances, you can refer to the files available on the website for the complete source code.

Part I

Professional WordPress Development

Chapter 1 WordPress As a Professional Web Development Tool

Chapter 2 Kicking Off a WordPress Project

Chapter 1

WordPress As a Professional Web Development Tool

WordPress was originally established as a tool for bloggers, aimed at the growing community of people writing personal, technical, or business blogs, and providing them with a platform they could use to host that blog on their own server. But WordPress has evolved—significantly. It is no longer simply a blogging tool, but a fully fledged content management system (CMS), with a myriad of features that enable developers to experiment with the structure and functionality of a site, customize the dashboard and admin screens for users, and install plugins to enable whatever additional capabilities the site needs.

WordPress, to put it simply, is now a professional web development tool, used by thousands of web professionals to build sites for themselves, their clients, and other users. It's a tool on which you can build a business.

This chapter looks at the WordPress features you can harness as a professional web developer, and identifies how your working practices may need to change if you're scaling up your WordPress practice. You'll learn some techniques for improving your working and coding practices when collaborating as part of a larger team, and find out how to manage large web design and development projects, including the skills you'll need and the people you can expect to work with. You'll also look at the implications of building and possibly selling WordPress themes and plugins for release to other users and developers.

What It Means to Be a Professional WordPress Developer

If you're reading this book, there's a good chance that you already use WordPress on a professional or semi-professional basis. Maybe you work for an agency that builds client sites in WordPress, or for a company with a WordPress site that you maintain. You could be a freelance WordPress developer, or perhaps you're starting out as a fully fledged WordPress professional, setting up your own agency and building WordPress-powered sites for your own clients.

If you're going to do this professionally, you'll have to adapt your working style and practices, as well as your approach to development and coding. As a bare minimum, you'll need to do the following:

- Ensure that you understand WordPress well enough to build a diverse range of complex sites with it.
- Change the way you code so that people you're working with can understand what you've done and work with your code.

- Start thinking imaginatively about WordPress development, and in particular about how you can harness WordPress to solve your clients' real-world problems.
- Develop the skills needed to explain to your clients how WordPress, and the site you design and develop using it, can benefit them.
- Come to grips with the more commercial aspects of WordPress—using it to enhance your clients' SEO, seeing the potential to maximize your earning potential from WordPress, and possibly start selling themes or plugins.

As you work through this book you'll see that there is no one-size-fits-all approach to being a WordPress professional, but there are some practices and capabilities all WordPress professionals need, and others that will be more relevant to you depending on exactly how you work with WordPress. We'll be revisiting this theme throughout the book, particularly in Part IV, "Pushing the Limits: The Best Tools for Site Development."

Professional Coding Practices

The first thing you need to do if you want to scale up your approach to WordPress is review the way you code. Ask yourself: Who else looks at or works with your code at the moment?

The answer will vary according to where and how you work. If you're not running your own agency, you are probably not the person with ultimate responsibility for the quality and robustness of your code. Conversely, you might be the only person who works with your code.

Professional coding practices are about more than writing valid, standards-compliant code, although that is essential—and hopefully you already do this. It's about writing code that other developers can happily work with and develop further. If you're developing themes or plugins for other WordPress users to install, then you may need to focus on writing code that is resistant to the kind of hacking a WordPress novice might subject it to.

There are a few aspects to professional coding practices:

- Make sure your code is valid and standards-compliant.
- Use up-to-date coding methods.
- Comply with the WordPress coding standards.
- Make your code tidy.
- Structure your files well.
- Be consistent.
- Use comments liberally.

The following sections describe what these guidelines mean in practice.

Valid and Standards-Compliant Code

Yes, I've already said this, but it is absolutely fundamental. If you haven't run the HTML in your themes or plugins through a validator, do it! The most popular method for validating your code is to use the W3C validator at

<http://validator.w3.org>. This is the most widely used approach and the first place to start. However, validating your code involves more than just this. It includes (but is by no means limited to) the following:

- Validating against accessibility standards, including WAI, or the Web Accessibility Initiative
- Checking links
- Validating feeds
- Cross-browser compatibility checking (including handheld and tablet devices)

For a long list of validation tools and techniques, take a look at the guidance on the WordPress codex at http://codex.wordpress.org/Validating_a_Website.

Up-to-Date Coding Methods

If others are going to be working with your code, especially if they're going to be paying for it, it's imperative that you write code that is up to date. For example:

- Don't use tables for layout (we really hope you stopped doing this a while back, but it bears repeating).
- Use the most recent versions of the main coding languages—HTML5 and CSS3.
- Avoid using deprecated code—although browsers are generally forgiving, your users may not be.
- Accept that you can't keep up to date with everything, but make sure you read web development blogs, journals, and magazines so you're not completely out of the loop.
- If a project involves something you haven't done for a while (or at all), do some research before starting—or hire a specialist as a freelancer or staff member.

WordPress Coding Standards

The WordPress codex details a set of standard coding practices, designed to help enhance consistency in WordPress code structure. This includes standards for PHP, HTML, and CSS.

Get to know these standards and use them. Even if you come across code that doesn't adhere to them, it's good practice to use them yourself and to expect members of your team to do so. The consistency and clarity that this brings to your code will help others who work with it, including your team—and your clients if you are selling themes or plugins.

Tidy Code

If other people are going to be working with your code, especially if they aren't advanced developers themselves, your code has to be easy to understand. Adhering to the following best practices will result in code that is easier to work with and harder to break:

- Use line breaks and indentation to help others see how your code is structured in one glance.
- Avoid empty divs and other elements added purely for styling—try to keep your markup lean and use CSS to style it, including the use of CSS pseudo-elements where appropriate.
- Rationalize your stylesheets to avoid duplication—if two or more elements or classes have the same styling, code it once instead of doing it repeatedly for each one.

Well-Structured Files

Files that have a clear structure are much easier for other developers to work with.

Your markup should be written in the order it appears on the page, even where you're using CSS to position it outside that flow. So if your layout shows the header first, then the content and sidebar followed by a footer, code it in that order. This will not only help other developers, but also improve accessibility, as screen readers read the code in the order in which it's written.

Structure your stylesheets in sections, with a summary of the structure at the beginning. For example, the Twenty Eleven theme (<http://wordpress.org/extend/themes/twentyeleven>) is split into no less than 21 sections, which include the following:

- CSS reset (this should always come first)
- Structure (for overall page layout)
- Global (for global elements such as fonts, list styling, and colors)
- Header (specific styling for the header layout)
- Menu (for the navigation menu)
- Content (for page or post content)
- Link (for links, including links in text and page titles)
- Image (for images within the content)
- Widgets
- Footer
- Media queries

Your themes may not need so many sections, but if you are developing themes for others to download and use, you'll need to cover all the bases.

Consistency

People who need to work with your code don't want any surprises, so consistency is essential.

For example, when coding HTML within PHP files, use a consistent method.

Some developers prefer to write opening and closing PHP tags every time they need to add markup, as shown in the following very simplified extract from the WordPress loop:

```
<?php if ( have_posts() ) while ( have_posts() ) : the_post(); ?>
  <article>
    <?php the_content(); ?>
  </article>
<?php endwhile; ?>
```

Other developers prefer to stay in PHP, and use `echo` instead:

```
<?php if ( have_posts() ) while ( have_posts() ) : the_post(); ?>
    echo "<article>";
    the_content();
    echo "</article>";
endwhile; ?>
```

The first approach is probably the most commonly used, and it is preferable if the people who will be working with your code are more familiar with HTML than they are with PHP. Whichever you choose, make sure you stick with it.

You should also use and format comments consistently. For example, you might prefer to add comments to your PHP on the same line as the code:

```
<?php the_content(); ?> // this is a comment
```

Conversely, you might prefer adding comments on separate lines:

```
<?php
/* this is a multiline comment
which spans more than one line */
?>
```

Again, whichever you do, be consistent and use the correct syntax.

CSS styles comments in the same way. HTML, however, uses only one kind of commenting syntax:

```
<article><!-- comments here --></article>
```

You already know how you prefer to code, and don't need us to tell you how to do it—but it's important to consider your preferences when other people will be working with that code, to ensure consistency and ease of understanding. You may also want to specify some coding practices and habits for other members of your team so you're all working in the same way.

Avoiding Duplication of Function and File Names

Whenever you write a new function, it's essential that you give that function a unique name to void any conflicts with WordPress functions or functions in other plugins or the active theme. It's normal practice to add a prefix to each function that corresponds to the name of your theme or plugin.

A function such as the following runs the risk of not being unique:

```
<?php
function name_of_function(){
    // function contents
}
?>
```

Instead, it should have a prefix, as follows:

```
<?php
function prefix_name_of_function(){
    // function contents
}
?>
```

In this book I prefix all functions with `wpptl_`:

```
<?php
function wpptl_name_of_function(){
    // function contents
}
?>
```

This also applies to any the names of any hooks you register and of files you create in plugins. It doesn't apply to theme template files—these need to adopt the standard filenames for template files if they're going to work.

Liberal Use of Comments

Still on the topic of comments, it's worth stressing how important they are when working with other developers. Try to anticipate any places in your code where a comment could be useful to someone unfamiliar with it. This will help other developers understand at a glance what your code is designed to achieve.

As I discuss later in this book, there is almost always more than one way to achieve something in WordPress, and in code. The next person to work on your files may have a different way they prefer to code something, so what you've done may not make sense to them. By adding a comment, you're telling them what your code does and where it applies.

Commenting liberally will also help you when you revisit some code you wrote a year ago on a client site that now needs updating. The chances are good that this will happen at 4:00 p.m. on a Friday when the client's site has hit a problem and you need to edit some offending code fast. Good commenting can mean the difference between finding the problem straightaway, making the edit, and being home in time for the weekend, versus slogging through thousands of lines of code and missing your kid's soccer game.

Following are some tips for good comments:

- Use comments at the beginning of your file to explain what the file does. For example, in template files include a note about what data it displays and any customizations you've made to the loop or other parts of the file; and in plugin files add a note regarding its functionality. For example:

```
/**
 * Template Name:      sidebar-products.php
 *
 * The include file used for the sidebar on pages using the
 * single-product.php template. Displays a gallery of product images
 * (omitting the featured image which is displayed in the content).
 *
 * @package WordPress Pushing the Limits
 * @since 1.0
 */
```

This comment tells users the name of the template file, what it does, the theme it is part of (`@package`), and which version of the theme it has been in place since (`@since`). You should use a similar system for plugin files.

- Use comments to demarcate the sections of your code, not only in stylesheets but also in your theme template files and functions file.
- Comment anything that is nonstandard.
- Comment anything that took you a while to work out—use detailed comments so when you come back to it, you don't have to think it through again.
- Comment anything that you know someone else will be working on—for example, if your theme files contain scripts that you'll be asking a JavaScript developer to perfect, add comments explaining where they apply in the site.
- Use wording in your comments that you can find using a search later—so don't abbreviate, and use terms others would understand.
- Whenever you comment out some code, add a comment to yourself containing the reason. That way, if you forget to remove the code after you've finished, or want to add it back in the future, you'll know what's going on.
- When in doubt, add a comment!

Approaching Large-Scale Site Design

So, now that you're running a professional WordPress business (or planning to), you're going to want to attract big clients—clients with exciting, complex requirements for their website, and the budget to match.

The reality, of course, is that not all of your clients will be like this—indeed, if you're new to this game, the chances are good that smaller clients will be your bread and butter in the early years, but it's a good idea to set your sights high and develop working practices in line with this goal.

You may have worked on large, complex sites before, perhaps as an employee in an organization's web team. However, you may not have had overall responsibility for the site, for managing its development, and for working with its stakeholders.

To work with clients and manage the completion of large-scale web design projects, you need some key skills:

- The ability to work with a range of stakeholders
- A balance between technical capability and business sense
- Communication skills
- Planning and project management skills
- Problem-solving skills

The following sections take a look at how each of these skills applies to large-scale site development.

Working with a Range of Stakeholders

When managing large-site builds, you typically work with a range of stakeholders who have hugely varying expectations and different levels of understanding regarding what you're trying to achieve. Some stakeholders you might need to work with include the following:

- **The client's project manager or website manager.** This may well be the person who commissioned the work (but may not have approved it). He or she will have a sufficient level of technical capability but may know little or nothing about WordPress, especially if the client is moving to WordPress for the first time. This individual is an important ally for you as someone with an interest in the project's successful completion but may not have the authority to make important decisions.
- **The client's senior managers or managing director.** These people will be less closely involved in the project but have the power to make key decisions, including pulling the plug on your project or drastically changing its direction. Not involving these individuals at key stages in the project can be a big risk—if you complete most of the work before you present the website to them and they don't like it, you may have to start again from scratch.
- **The client's technical staff or website editors.** The client may have employees whose job it will be to maintain and edit the site once it is launched—unless they require you to do this. If not, you may have to train these employees to use WordPress and/or work collaboratively with them on some of the code. They are unlikely to be decision-makers.
- **Your own team.** You may have one or more partners in your agency, and a team of developers and designers, either freelance or employed by you. Your role is to ensure that they know their role in any project you're managing, they have the information and resources they need to do the work, they get feedback on that work, and they know what the deadlines are. You also need to oversee the quality of their work and intervene if it is not up to scratch.

As you can see, you could be working with some very different people; therefore, in order to ensure that your projects are successful, you'll likely have to adopt a different approach with each group—something you may never have done before if you're used to spending most of your time on development. You'll need to use the set of skills described in the following sections to make it work.

Balancing Technical Capability and Business Sense

Technical capability is essential. If you weren't already proficient in WordPress, you wouldn't be setting up a WordPress business. But you also need business sense, which comes into play not only when you're dealing with clients, but also when you're managing your own business. Obviously, it's beyond the scope of this book to teach you how to manage your accounts, form a business plan, and so on—you can find hundreds, if not thousands, of resources to help you with that—but it's worth noting that having or developing a business sense will help you choose the right WordPress projects and make a success of them.

When you're talking to a client about their website build, it's important to be able to step back from the details of a project to see the bigger picture—so you can translate technical knowledge into something that has a tangible benefit. For example, you may be great at building responsive sites, but if you can't explain the benefits to your client, they won't pay you to do it.

Communication Skills

"Code is poetry," or so the saying goes, and it is—well-written code can make a website soar, in terms of content, functionality, and performance. However, most clients don't understand code—and they probably don't