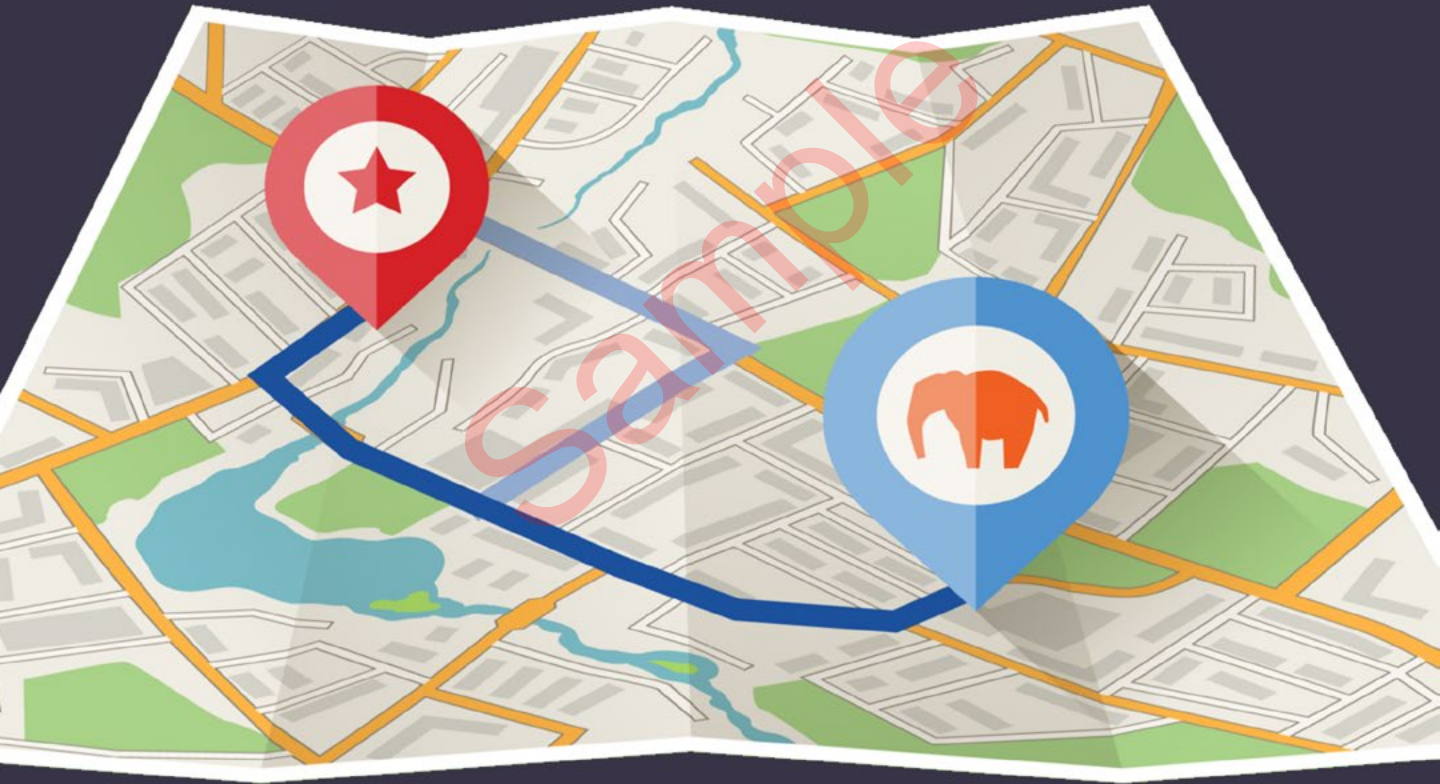


What's Next?

Professional Development Advice

from php[architect] magazine



Edited By Oscar Merida

 a php[architect] anthology

What's Next? Professional Development Advice

From php[architect] Magazine

Edited By
Oscar Merida

Table of Contents

Contributors	IX
Cal Evans	IX
Joe Devon	IX
Steve Grunwell	IX
Stefan Koopmanschap	IX
Ole Michaelis	IX
Joshua Silver	IX
Gabriel Somoza	IX
David Stockton	X
Jordan Tway	X
Joshua Warren	X
Eli White	X
Trip Witt	X
Chapter 1. Professional Development for Professional Developers	1
Programming Should be Your Passion	2
Keep Up with the Industry...	2
Don't Worry About What Other People are Doing	3
Keep an Eye on Your Stress Levels	3

TABLE OF CONTENTS

Keep a Record of Your Ideas	4
Open-Source FTW	4
Code Speaks Louder than Words	5
Never Stop Learning	5
Find (and be) a Mentor	6
Learn About Humility	6
Be a Positive Force in Your Community	7
We're All in This Together	7
Chapter 2. More Than Just an “OK” Dev	9
1. Engages with the Community	10
2. Contributes to Open Source	10
3. Learns from Mentors	12
4. Is a Full-Stack Integrator	13
5. Follows Coding Standards and Best Practices	14
Conclusion	15
Chapter 3. How to Burn the Candle at Both Ends	17
A Personal Story	18
Changing the Culture	18
What Was the Point?	19
The Story of Face Punch Day	20
Where We Are Today	21
Burnout and YouMe	21
Conclusion	23
Chapter 4. You Had One Job	25
Why This Topic?	25
What Does a Senior Developer Do?	26
Software Engineering Job Levels and Responsibilities	27
Software Development Apprentice	27
Software Developer	28

Software Engineer	28
Senior Software Engineer	28
Staff Software Engineer	28
Principal Software Engineer	29
Beyond Senior	29
With Great Power...	30
How Can You Do This?	30
It's About the Mentoring	31
Climbing By Pulling Others Up	32
Closing	32
Chapter 5. Getting the Most out of a Conference	33
Know What You Want to Accomplish by Attending	34
Review the Schedule and Select the Must See Sessions	34
Make a List of the People You Want to Meet	34
Do Not Make Plans for Every Session	35
Don't Overdo the After Events	35
Take and Share Notes	36
Make Time for Yourself	36
Chapter 6. PHP Conference Newbies 101	37
10 Ways to Maximize Your First Experience	37
Lesson 1: You Are Neither the Dumbest nor the Smartest Person in the Room	38
Lesson 2: Speak Up	38
Lesson 3: How to Get a Free Pound of Bacon—Meeting the Conference Organizers and Speakers	38
Lesson 4: Here, There, and Everywhere—Attending Events and Making the Most of the Schedule	39
Lesson 5: Be Prepared	39
Lesson 6: Know Your Frameworks...	40
Lesson 7: Know Your Bartender—How to Get Around in the City	40
Lesson 8: Sponsors	40

Lesson 9: Leave Work at Work	41
Lesson 10: Come Back—the Return of the Attendee	41
Side Bar—from the Boss	42
Chapter 7. Fear of Public Speaking	43
Face the Fear	44
Invitation to Speak on a Panel	44
Impostor Syndrome	45
Are These Just Words?	45
Chewing Gum and Other Tricks	45
Culmination of a Dream	45
What Was the Talk About?	46
Spreading Awareness of Accessibility	46
What Else Happened at Midwest PHP?	47
Chapter 8. Two Years of PHPAmersfoort: Running a Local Usergroup	49
Our Start	50
Preparing a Plan	51
Tools We Use	51
Funding the Group	54
Finding Venues	55
Getting Speakers	55
Meet-up Logistics	56
Special Activities	57
The Crew	57
Starting Your Own Usergroup	57
Chapter 9. Finding the Solution to the Problem	59
Request for Changes	60
PDF Combinatorics	60
The Iron Triangle of Software Development	62
Why Do We Build Software?	62

Make it Work, Make it Right, Make it Fast	63
Why Being DRY is Not Always Best	64
Conclusion	64
Chapter 10. Introduction to Product Management for Developers	65
Defining Feature Scope, Timeline, and Priorities	66
Understanding the “Why”	68
Recommended Reading	70
Next Steps	70
Chapter 11. Choose Your Own Adventure—Freelancer or Founder?	71
Starting Out	71
The Pros and Cons of Freelancing	72
The Pros and Cons of Owning a Development Agency	72
How to Choose	73
Building a Freelance Career	75
Building a Development Agency	76
Chapter 12. On the Value of a Degree...	79
Chapter 13. Resume 101	81
Things I Look For in a Candidate When Recruiting	84
Chapter 14. Finding Your Voice	87
Step 1: Finding Your Topic	87
Step 2: The Abstract	88
Step 3: Name the Talk!	89
You’re Done	89
Permissions	91
Index	92

Sample

Contributors

Cal Evans

These days, when not working with PHP, Cal can be found working on a variety of projects like Nomad PHP. He speaks at conferences around the world on topics ranging from technical talks to motivational talks for developers. [@calevans](#)

Joe Devon

Joe Devon has been developing in PHP since Version 3. He is an advisor or board member at various companies, non-profits and conferences. He is Founding Partner of LAPHIP user group; Diamond, a Digital Agency in Venice; Television Four, a Digital Media startup and Global Accessibility Awareness Day (#GAAD). He is passionate about Digital Accessibility & encouraging the UX & Development Community to produce accessible content. He blogs at <http://dws.la/author/joedevon> and may be reached on LinkedIn, <http://linkedin.com/in/joedevon>, Twitter [@joedevon](#) and email at phparch@dws.la.

Steve Grunwell

Steve Grunwell is a Senior Web Engineer at 10up, living and working in Columbus, OH. When he's not writing software, he can be found speaking at conferences, blogging about software development, or continuing his search for the perfect cup of coffee. [@stevegrunwell](#)

Stefan Koopmanschap

Stefan Koopmanschap runs his own business called Ingewikkeld, organizes WeCamp, and has been involved in starting several usergroups in The Netherlands and BeNeLux. [@skoop](#)

Ole Michaelis

Ole Michaelis is the co-founder of SoCoded, a hackfest and web development conference in Hamburg. He is a Software Engineer at Jimdo, a DIY website creator, where he focuses on backend development. In his free time, he's building slidr.io, a hassle-free slide sharing platform. Ole is passionate about open source software, enjoys traveling, and loves Mexican food. He classifies himself as a 'bad' German as he dislikes beer and soccer – the traditional German past-times. [@CodeStars](#)

Joshua Silver

Joshua Silver is VP, Product Development and the technical co-founder of Patientco, a healthcare IT and payments startup in Atlanta. At Patientco, he is responsible for overall product direction, R&D initiatives, and technology strategy. Joshua has over a decade of development (mostly PHP/MySQL) and product management experience and frequently speaks on topics such as consumer-centric system design, secure payment processing, and scalable web architecture. Joshua holds a BS in Computer Science from GA Tech. When not working, Joshua is most likely traveling and has been to nearly 30 countries.

Gabriel Somoza

Gabriel Somoza is a Belgium-based PHP architect with a strong background in Ecommerce. He spends most of his time consulting through Stratergy—his development boutique specializing in building highly-customized web applications and e-commerce stores. He's also a *ZCE* and *Magento Certified Developer (Plus)*. He is active in the open-source community with contributions to Doctrine Migrations, ZF2, and is the author of the Stratergy InfiniteScroll extension for Magento. He's the organizer of PHP Limburg, a meetup with 120+ registered members in Belgium. When not working,

CONTRIBUTORS

he spends time with his beautiful wife, travels, and researches new ways of applying technology to solve old problems. [@gabriel_somoza](#)

David Stockton

David Stockton is a husband, father and Software Developer. He builds software in Colorado, leading a few teams of software developers creating a very diverse array of web applications. His two daughters, age 11 and 9, are learning to code JavaScript, Python, Scratch, a bit of Java and PHP as well as building electrical circuits, and a 4 year old son who has been seen studying calculus, equating monads to monoids in the category of endofunctors, and is excelling at annoying his sisters. David is a conference speaker and an active proponent of TDD, APIs and elegant PHP. He's on twitter as [@dstockto](#) and can be reached by email at levelingup@davidstockton.com.

Jordan Tway

Jordan Tway is a lead recruiter at Robert Half Technology. In his free time, he loves to stay active! That includes lifting weights, playing basketball, soccer, sand volleyball, golf, flag football, and he is currently training for the Ironman 70.3 Triathlon in Racine. On Wisconsin and Go Pack Go! Outside of staying active, he likes to meet new people, listen to music, hang with friends in the area, and relax. Before Robert Half, he was enrolled at the University of Wisconsin-Whitewater studying Marketing with an emphasis in Direct and Internet Marketing. He always enjoys connecting with people so feel free to look him up on LinkedIn or follow him on Twitter! [@JTway23](#)

Joshua Warren

Joshua Warren has been developing with PHP since 1999. He is the Founder of Creatuity, where he works on eCommerce projects with Magento and Yii. He founded Creatuity in 2008 as a development agency and has grown it to a team of over 25 including 14 developers working across 2 continents. When Joshua isn't developing, he enjoys traveling to new countries to try local food and beer while meeting developers at local meetups. You can follow him on Twitter or on his blog <http://joshuawarren.com>

Eli White

Eli White is the Conference Chair for php[architect] and Vice President of One for All Events, LLC. He has a BS in CS, but it was so long ago he probably doesn't remember any of it at this point. [@EliW](#)

Trip Witt

Trip Witt is a web application developer with ten years experience living in Lynchburg, Virginia. He works for Advanced Manufacturing Technology, Inc. creating and maintaining information systems and custom web applications both internal and client-specific. When he was just 27 he was named one of Central Virginia's Top Movers and Shakers for his website development with the American Red Cross. Customer focused and driven to exceed expectations, he is always wanting to learn more and thrives to develop new applications to make life easier for the end user. [@trip_witt](#)

Chapter 2

More Than Just an “OK” Dev

Gabriel Somoza

In today’s web development market there are tons of job openings with excellent salaries and perks like cars, work-from-home arrangements, unlimited holidays, opportunities to become certified, and so on. But how long until the bubble bursts? A growing number of people are being raised to become programmers, and some have been programming since before they could even talk. Are you ready to compete with the upcoming waves of younger, smarter developers?

I was raised in Argentina, one of a few countries where you wouldn’t necessarily be surprised to learn that your taxi driver had a medical degree. As in many other parts of the world, doctors in Argentina are overworked and underpaid.

At the other end of the spectrum, we developers experience almost the exact opposite. Companies are competing very aggressively to attract talent, to the point of basically treating developers like rockstars—giving them beautiful company cars, unlimited holidays, remote work arrangements, and even four-day work weeks, all on top of an average U.S. salary of nearly six figures—and well above that in some other places.

When considering the contrasting working conditions of these two professions, one thing is

MORE THAN JUST AN “OK” DEV

clear: we’re living in the golden age of web development. In the words of Silicon Valley developer James Somers^[1]: “In this particular gold rush (of tech startups) the shovel is me.”

Such market conditions can foster developer complacency and professional stagnation. For example, a developer might stay long-term at a comfortable and well-paid job that’s not challenging enough for his or her career, and could simply lose the drive to learn new things.

I regularly hire developers and frequently encounter applicants who seem content to stagnate. As I interviewed people, I learned to quickly identify certain characteristics that, in most cases, make all the difference between someone being an “OK” developer and an outstanding one. I have listed some of them below, along with some advice on how to build them into your career toolbox.

1. Engages with the Community

This refers to developers who regularly attend meetups or conferences and, usually, are active on IRC, Slack, and other online forums.

Ideally, the developer is a community **enabler**: actively helps organize events, regularly speaks at meetups or conferences, and provides support in online technical communities. A community enabler can communicate and solve complex interpersonal problems beyond the realm of code.

These traits are valuable because communities are bigger than individuals. A developer actively involved in the community is connected to something larger than himself and will therefore have continual access to:

1. New ideas, which can prevent stagnation.
2. Other experts and specialists, including the problem-solving resources and potential new hires they can supply. Such value is often overlooked.
3. Mentors.

Starting to Participate in the Community

The development community will gladly help you to integrate if you show interest. You can start by joining a local meetup (try <http://meetup.com>). Twitter, IRC, and Slack are great venues when you’re having trouble or if you’re willing to grow by helping others.

2. Contributes to Open Source

Open source is the heart of PHP, and I’ve come to realize that great PHP developers understand this. Contributing to Open Source projects is therefore one of the best ways to give back to the community.

I now rarely consider hiring someone for a senior position if that person doesn’t have publicly

[1] *Web Developer: It’s A Little Insane How Well-Paid And In-Demand I AM:*
<http://phpa.me/bi-webdev-demand>

available contributions, even just a personal project. The project must be complete, and, ideally, it should be a reusable library that solves a well-defined problem.

When reviewing a developer's public code, it's important to look for the basic indicators of a good developer: automated tests, a well-defined coding style, documentation, design, license inclusion, proper use of patterns, proper use of Composer (including a good choice of dependencies), and so forth.

A developer who regularly contributes to open source displays the following characteristics:

1. experience designing code for reusability
2. self-motivation and proactivity
3. attention to detail

Other ways of giving back to the community, such as mentoring, will be discussed later in this article .

Starting to Contribute to Open Source

When I wanted to start contributing, many people told me that I simply had to submit a pull request. My problem was not knowing *what* to contribute. Eventually, I realized that getting started is not so much about chasing a great idea as it is about coding with a specific *mindset* of solidarity: "Will this code I'm writing be valuable to the community?" The answer is probably **yes** if you:

1. Fixed a bug
2. Found a better way of doing things (something faster, less resource-intensive, more flexible, etc.)
3. Found something that your framework was missing. For example, one of my first pull requests was a Whitelist and Blacklist filter for Zend Framework 2.
4. Created something modular that can be reused with a framework, or even by itself.
5. Helped to document a framework or library
6. Added automated tests for a framework or library

As you explore the mindset and become familiar with the above, you'll find many other opportunities to contribute code, and ultimately give back to the community.

TIP #1: *Before spending a lot of time working on a new feature or fixing a bug on a project, go to the project's issue tracker and make sure there are no open issues related to that.*

TIP #2: *If you're contributing a new feature or a backwards-incompatible change, make sure you open a new ticket on the project's issue tracker to discuss it first, before you spend time working on it. The project maintainer may have different plans in mind, so first make sure your efforts will be well received!*

3. Learns from Mentors

A mentor is a person or friend who guides a less experienced person by building trust and modeling positive behaviors. An effective mentor understands that his or her role is to be dependable, engaged, authentic, and tuned into the needs of the mentee^[2].

It's relatively common for people to find a mentor because it's natural for human beings to transmit knowledge from generation to generation. Many people have one (or more) mentors during their careers without even realizing it: a teacher at school, a family member, a colleague or boss, or even a consultant working at their company.

When I look back I realize my career wouldn't be the same if it wasn't for the mentors I had.

For me, it started early: when I was around 11 years old my computer teacher saw that I had an interest in programming. While most other kids were learning how to type without looking at the keyboard, he would invest time to show me how to build a few things in Macromedia Flash 5 (ActionScript). That's how I started to build my first Flash games and websites, and how I developed the passion for web development that I still have today. Since then I've been lucky to always have at least one mentor in my life. Eventually it became a conscious choice: I started to actively look for mentors to coach me on several aspects of my career.

One of the main reasons I search for developers who have mentors is that I know I'll be hiring someone who has direct access to a pool of knowledge and experience that's larger than the individual. This is similar to the concept of being tapped into a community, except that in this case it carries the stronger bond of a deeper relationship.

Finding a Mentor

There are many paths to finding a good mentor, but in my experience the best mentoring relationships happen **organically**. The most important ingredients that help these organic relationships to develop are being involved in the community and giving back to it—which, not by coincidence, are the first two **items** discussed in this list. While doing that you'll cross paths with people that you'll start to admire.

Once you find the right person, the next step is formalizing the relationship. There are many articles online on how to engage with a potential mentor and ultimately define and formalize the relationship.

Last but not least, in the PHP community there's a project at <http://php-mentoring.org> specifically designed to help people find mentors. The site also has more information about the benefits of mentoring as well as a directory of people who are willing to mentor. I encourage you to check it out!

[2] *Module 3: What is a Mentor and Roles of the Mentor and Mentee:*
<http://www.oycp.com/MentorTraining/3/m3.html>

Index

A

abstraction, 64
agile
 environment, 67
 software development
practices, 61
anxiety, 2, 4, 39, 43–46
architect, 27, 31, 46–47
attendees
 first-time, 38
 professional conference, 33
 recruiting, 55

B

boss, 6, 12, 34, 36, 42, 54
budget, 3, 73–75
bugs, 5, 11, 23, 27, 43, 88
 fixing, 80
 good reporting, 5
 last, 66
 new, 59
 tracker, 19
 uncaught, 69
burnout, 3–4, 19, 21–23, 31
 syndrome, 23
bus factor, 31
business plan, 73, 75–76

C

callbackwomen, 89
Call for Papers (CfP), 39, 87, 89
callingallpapers.com, 89
career
 growth, 82
 path, 29

 transition, 70
caregivers, 61
clients, 53, 69, 72–73, 75–76, 78, 85
 freelance, 76
code
 designing, 11
 production, 80
 public, 11, 13
 refactor, 66
 review, 14
 standards, 14
code analysis, static, 15
Code Katas, 5
CodeSniffer, 15
coding standards and best practices, 14
community
 involvement, 75
 resources, 4
 technical, 10
compensation, 56, 86
Computer Engineering, 79
Computer Science, 31, 79–80, 89
conferences, 2–3, 10, 32–34, 36, 38–42, 47, 50, 52, 56, 75, 87–89
 attending, 41
 organizers, 38–39, 52
 PHP, 33, 37–38, 40
contracts, 73, 81, 84
contract-to-hire, 81, 84
contract work, 84
culture, 18–20, 23, 69

D

death marches, 19, 31
degree, 22, 31, 79–80, 82
dependency injection, 63
depression, 4, 22–23
developer
 community, 38
 core, 46–47
 full-stack, 13
 full-stack integrator, 13–14
 good, 11, 68
 jobs, 85
 junior, 6, 27, 30
 open source web applica-
 tion, 42
 senior level, 32
developers, senior, 25–26, 29–30, 32
development agency, 72–78, 83
development methodology, 83
development team, 13, 18, 69

E

education, 40–41, 82
employers, 3–4, 29, 81–82, 84–85
estimates, 18, 66
evaluations, technical, 84, 88
Evernote, 4

F

Facebook, 40, 47
family, 3, 22, 44, 73, 75
fear, 2, 43–46

feedback, 5, 23, 52, 55, 66,
72, 85
freelance, developers, 71–73,
75–78
freelancer, 71–72, 74–76, 78
 career, 73–75
 lone, 72

G

GitHub, 5, 75
group chat, 53

H

hiring, 10, 12, 79, 82, 84–85
 focus, 77
 manager, 80, 82

I

Impostor Syndrome, 19, 45,
80
internals, 5–6
interview, 14, 18, 26, 81–83,
85
 preparation, 86
IRC, 10, 75

J

job titles, 26
joind.in, 52, 55

K

Kanban-board, 53

L

LinkedIn, 82

M

management, 19, 29, 79

marketing, 66, 73, 77
meetups, 10, 47, 51, 54
 local, 10
 organizing, 44
mentors, 5–6, 10, 12–13,
29–30, 32
 effective, 12
 formal, 6
mentor women, 41

O

open source
 code, 14
 framework, 46
open-source, software, 4–5,
62
Open Sourcing Mental
Illness, 4

P

PHP community
 international, 55
 local, 41
PHP-FIG, 14
phpmentoring.org, 6, 32
PHP Mess Detector, 15
PHPUnit, 50
PHP Women, 41
product management, 65–66,
68, 70
project manager, 18

Q

quality, 27–29, 31, 62, 65,
69, 76

R

recruiters, 5, 32, 55, 81,
84–86
 technical, 81

recruiting, 54, 84–85
refactor, 6, 63–64, 67–68
remote work arrangements, 9
requirements, 55, 68, 81
 correct, 62
 misunderstood, 61

S

Skype, 44, 53
Slack, 10, 52–53
speakers, 33–34, 36, 38–39,
44–45, 49, 51–58, 64, 87–88
 international, 56
 new, 55
 potential, 52
 regular, 55
 session, 40
speaking.io, 89
standards, 14–15, 26
Startup culture, 23

T

TDD, 19
 workflow, 63
technical debt, 67
 increasing, 29
training, 19, 31–32, 75
 budget, 32
 formal, 65, 80
Trello, 53
twitter, 2–3, 10, 19, 40, 72, 75

U

user groups, 2, 32, 41
users, 61, 67–70

W

workflows, 2, 88

php[architect] Books

The php[architect] series of books cover topics relevant to modern PHP programming. We offer our books in both print and digital formats. Print copy price includes free shipping to the US. Books sold digitally are available to you DRM-free in PDF, ePub, or Mobi formats for viewing on any device that supports these.

To view the complete selection of books and order a copy of your own, please visit: <http://phparch.com/books/>.

- **Functional Programing in PHP, 2nd Edition**
By: Simon Holywell
ISBN: 978-1940111469
- **Web Security 2016**
Edited by Oscar Merida
ISBN: 978-1940111414
- **Docker for Developers**
By Chris Tankersley
ISBN: 978-1940111360 (Print edition)
- **Building Exceptional Sites with WordPress & Thesis**
By Peter MacIntyre
ISBN: 978-1940111315
- **Integrating Web Services with OAuth and PHP**
By Matthew Frost
ISBN: 978-1940111261
- **Zend Framework 1 to 2 Migration Guide**
By Bart McLeod
ISBN: 978-1940111216
- **XML Parsing with PHP**
By John M. Stokes
ISBN: 978-1940111162
- **Zend PHP 5 Certification Study Guide, Third Edition**
By Davey Shafik with Ben Ramsey
ISBN: 978-1940111100
- **Mastering the SPL Library**
By Joshua Thijssen
ISBN: 978-1940111001