www.phparch.com

August 2021
Volume 20 - Issue 8

php[architect]

Trim One's Sails

Using Algebras in PHP

Boosting User Perceived Performance with Laravel Horizon

CLOUDWAYS

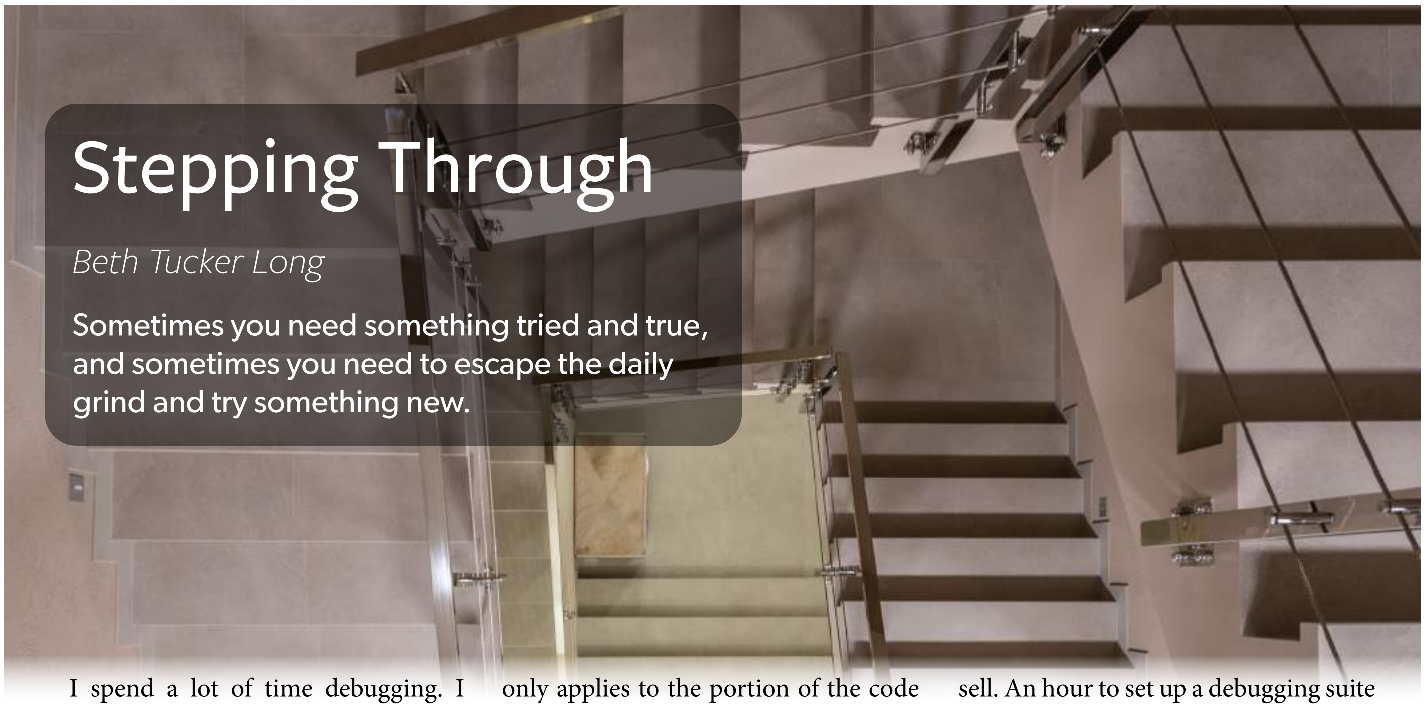
Moving Dreams Forward

Optimal PHP Hosting for Zero Downtime and Best Performance

Multiple performance tests show Cloudways improves **loading times for websites by 200%**! With innovative features like an **optimized stack**, advanced built-in caches, CloudwaysCDN, PHP 7.3 ready servers and so much more, Cloudways enables you to build apps with **unmatched performance** and **higher conversion rates.**

Promo: **PHPARCH**
20% off for 3 months

www.**cloudways**.com



Stepping Through

Beth Tucker Long

Sometimes you need something tried and true, and sometimes you need to escape the daily grind and try something new.

I spend a lot of time debugging. I mean, A. LOT. As a consultant, debugging is about 90% of my work, with the other 10% being meetings, or so it seems most days. For each new project my team and I start, the first phase is always "getting to know the codebase." We have found that the best way to get to know a codebase is to debug some small issues. Stepping through the code, tracing through the functions, figuring out who is passing what and how it changes is a massive help once we move on to quoting on new features or fixing larger items.

While I have heard the accolades of test-driven development for most of my career, I'll admit that it has never truly become my go-to. As a consultant, I rarely work in my codebase, and I often do not get to choose the tech stack we need to work within. Sure, I could just require that my customers pay fo9r the extra time for writing tests, automatically including it in my quotes. A bit more money up front will save them a lot down the road, but I can only require as much as the market will bear (and my kids insist on eating every day). While I have convinced a few clients to invest up front, many cannot, especially with the tenuous market we have had the past few years.

Even if I convince a client to move forward with good test coverage, that only applies to the portion of the code that I am adding or replacing, so it will take a while to get the tests in place. In the meantime, we still need debugging. As many features and benefits as the modern testing frameworks and tools offer, I find myself most often turning to old-fashioned step debugging. A couple of `var_dump()` calls, a die with a number returned to figure out how far the script is making it before it errors out—these tools are so simple to use that it is sometimes overwhelming to think of using anything else. Trying to weigh the timing cost of setting up a suite of debugging tools when I may be able to solve the issue in roughly the same amount of time. Then again, how many issues am I going to need to debug? An hour to set up a debugging suite versus an hour to solve the problem with manual debugging? Hard sell. An hour to set up a debugging suite versus an hour each when you have 20 different issues to debug? That's a much easier sell.

The trick is to take that broader view and look long-term. It's hard to get motivated about adding extra work to solve the one immediate problem in front of you. It's easy to just want to get the problem solved. Looking ahead, though, how much pain will you save yourself if you invest some time now to expand your available debugging tools? This is something I often forget. It turns out that investing in the long term is not just a tough sell for my clients, it's hard to convince myself of as well.

This month, I encourage all of you to try out one new debugging tool. Pop into the php[architect] Discord chat and let me know what you tried. Tag me and bug me about trying it too.

Beth Tucker Long is a PHP developer and organizer of the Madison Web Design & Development[1] and Full Stack Madison[2] user groups. You can find her on Twitter or on her blog[3]. Beth is a firm believer in promoting community and mentoring. She runs Treeline Design[4], a web development company, and Explioricon[5], a gaming convention, along with her husband, Chris. @e3BethT

1 *Madison Web Design & Development:* http://madwebdev.com

2 *Full Stack Madison:* http://www.fullstackmadison.com

3 *blog:* http://www.alittleofboth.com

4 *Treeline Design:* http://www.treelinedesign.com

5 *Exploricon:* http://www.exploricon.com