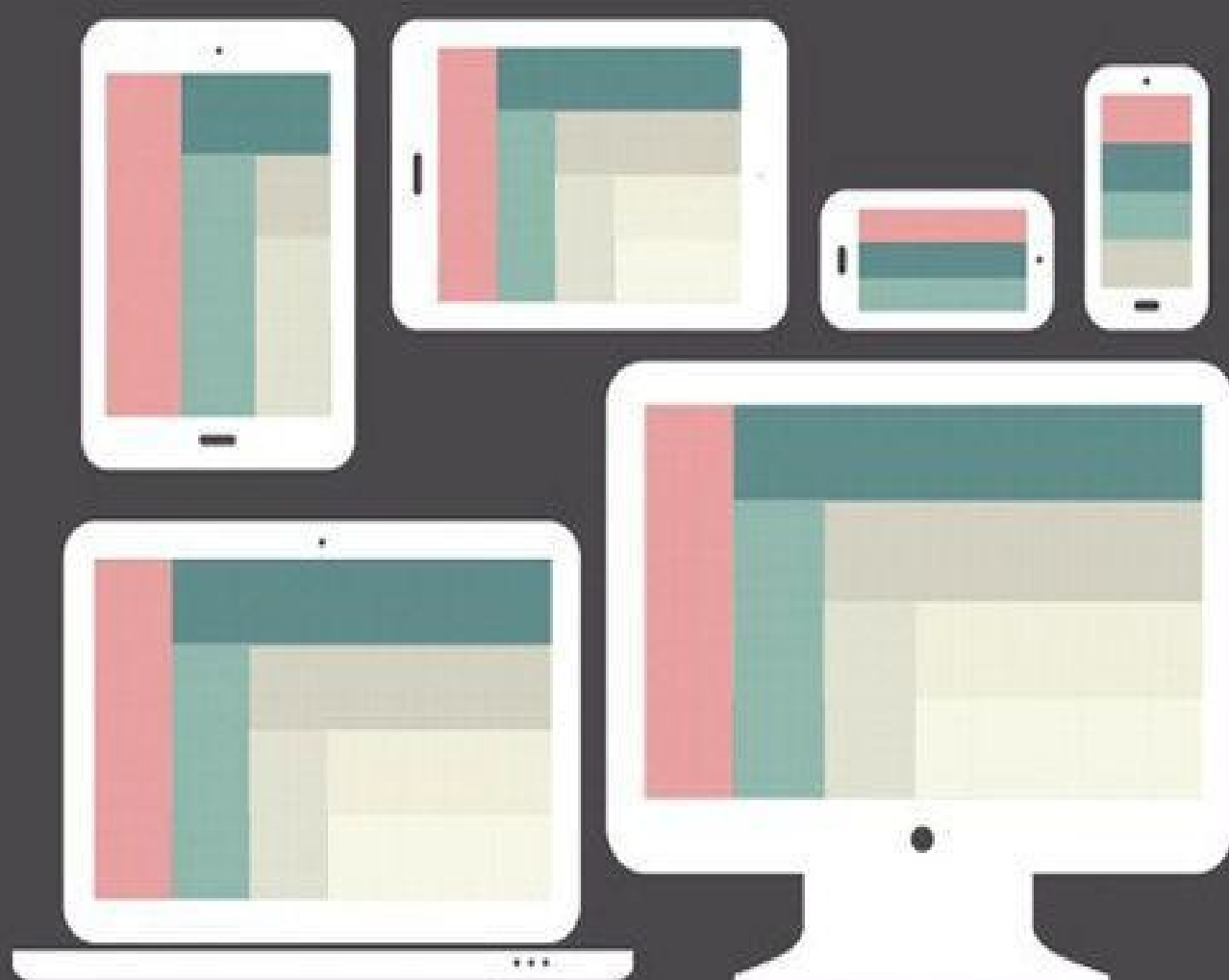


RESPONSIVE WEB DESIGN

with **Adobe Photoshop**

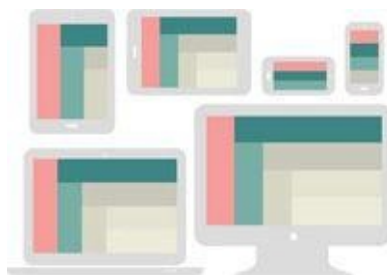


DAN ROSE



RESPONSIVE WEB DESIGN

with Adobe Photoshop



DAN ROSE



Responsive Web Design with Adobe Photoshop

Dan Rose

Copyright © 2015 Dan Rose

Adobe Press books are published by Peachpit, a division of Pearson Education.

For the latest on Adobe Press books, go to www.adobepress.com. To report errors, please send a note to errata@peachpit.com.

Adobe Press Editor: Victor Gavenda

Development Editor: Stephen Nathans-Kelly

Production Editor: Maureen Forys, Happenstance Type-O-Rama

Compositor: Cody Gates, Happenstance Type-O-Rama

Technical Editors: Dennis Kardys and Joel Baer

Copyeditor: Kim Wimpsett

Proofreader: Kim Wimpsett

Indexer: Jack Lewis

Cover Design: Aren Straiger

Interior Design: Maureen Forys, Happenstance Type-O-Rama

Notice of Rights

All rights reserved. No part of this book may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Notice of Liability

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author, Adobe Systems Incorporated, nor the publisher shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

Trademarks

Adobe, Photoshop, Fireworks, Dreamweaver, InDesign, Illustrator, Creative Cloud, Generator, Typekit, and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. All other trademarks are the property of their respective owners.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

Printed and bound in the United States of America

ISBN-13: 978-0-134-03563-5

ISBN-10: 0-134-03563-1

9 8 7 6 5 4 3 2 1

Acknowledgments

This book was guided by the resourcefulness of Victor Gavenda, the eagle eye of Stephen Nathans-Kelly (and the Adobe Press team), and the sage wisdom of Dennis Kardys. My gratitude to Dan Mall for trailblazing the Photoshop-for-RWD path. My love and dedication to Amanda for her support and encouragement, to Holly and Norah for play and snuggle breaks, and to Jesus, with whom all things are indeed possible.

Contents

1 Photoshop's New Groove

[Called Into Question](#)

[Stick in the Mud](#)

[*Fear of the Unknown?*](#)

[*Can I Still Get by Without Knowing Code?*](#)

[More Process Than Tool](#)

[A Battle of Two Short Words](#)

[Not on the Menu Tonight](#)

[*The Core Tenets of Responsive Web Design*](#)

[*Responsive Patterns*](#)

[*Performance*](#)

[*Photoshop Basics*](#)

[*The Minutiae of Version Disparity*](#)

[*The Merits of Comparable Tools*](#)

[Finding Photoshop's Groove](#)

[We Need to Make This Responsive!](#)

2 How Did We Get Here?

[How We Used to Know Photoshop](#)

[The Faults of Traditional Photoshop](#)

[*On Full-Page Comps*](#)

[Pain Point du Jour](#)

[*Fixed-Width Comps*](#)

[*Lack of Interactivity*](#)

[*Some Fonts Are Better Than No Fonts?*](#)

[*The Big Reveal*](#)

[*What Did You Expect?*](#)

[*Presentation Woes*](#)

[*Bound by Approval*](#)

[*Not So Stable*](#)

[*Less-Than-Seamless Exporting*](#)

[*Empty Your Pockets*](#)

[*Double the Effort, Double the Pain*](#)

[If Not Photoshop, What?](#)

3 The Case for Designing in the Browser

[You Get a Tool! And You Get a Tool! Everyone Gets a Tool!](#)

[Designing in the Browser 101](#)

[*Text Editor and Live Preview*](#)

[*Inspect Element*](#)

[Fluid by Nature: The Inherent Benefits of the Browser](#)

[*Interactivity*](#)

[*Global Changes*](#)

[*Free*](#)

[*1x the Effort*](#)

[Web Design's Natural Habitat](#)

[*Public Testing*](#)

[*PSDs for Proofreading, Browser for Evaluating Behavior*](#)

[*Reaffirming Expectations That Things Look Different in Different Browsers*](#)

[*Easy to Change on the Fly*](#)

[*Assessment as a Client Education Tool*](#)

[*Fold*](#)

[Designer/Developer Bonding](#)

[OK to Kill Photoshop Now?](#)

4 A Plea for Photoshop–Browser Harmony

[Photoshop Is the New Vinyl](#)

[The Power of Manipulation](#)

[Creative Mode vs. Correct Mode](#)

[The Path of Least Resistance](#)

[Responsive Design Sameness](#)

[Using Photoshop Only When Necessary](#)

[*The Megaman Principle*](#)

[*Practical Photoshopping: An Overview*](#)

5 Vetting Direction

[The Contrast Conundrum](#)

[*The Comp Approach*](#)

[*Within the Realm of Possibility*](#)

[Including Your Stakeholders in the Design Process](#)

[Moodboards](#)

[*Methods of Moodboarding*](#)

[*Finding and Storing Inspiration*](#)

[*Visual Inventories*](#)

[*The Pursuit of Efficiency*](#)

[*Conversations, Not Deliverables*](#)

[*Experimenting with Style*](#)

6 Establishing Style

[*Suitable Mock-up Replacements*](#)

[*On Sketching*](#)

[*Style Tiles*](#)

[*Style Prototypes*](#)

[*Component Inventory*](#)

[*Element Collages*](#)

[*Stripping Out the Abstraction*](#)

[*Crafting an Element Collage*](#)

[*Covering a Lot of Ground Quickly*](#)

[*Do Not Make It Look Like a Website*](#)

[*Color Comparisons*](#)

[*Scope Creep*](#)

[*Asking the Right Questions*](#)

[*Do Make It Look Like a Website*](#)

[*Point of Reference*](#)

[*I Still Can't See It*](#)

[*What's Missing*](#)

7 Establishing the System

[*Now It's the Browser's Turn*](#)

[*Defining the Style Guide*](#)

[*Web-Specific*](#)

[*Why the Style Guide Should Live in the Browser*](#)

[*Building the Component Library*](#)

[*Contents of a Comprehensive Component Library*](#)

[*Choosing the Best Environment for Your Components*](#)

[*Prototyping*](#)

[*Roughing It in Low-Fidelity*](#)

[*High-Fidelity and Beyond!*](#)

8 Getting Back into Photoshop with Page Layers

[*Rough Waters Ahead*](#)

[Introducing Page Layers](#)

[The Struggle to Increase Fidelity](#)

[Don't Get **Too** Comfortable in Photoshop](#)

[Leveraging Linked Smart Objects](#)

[There's No Easy Way to Suggest Tweaks](#)

[The Old Screenshot](#)

[The New Screenshot](#)

[Our Pages Lack Cohesion](#)

[Framing Content and the Big Picture](#)

[Where Skeuomorphism Worked](#)

[Some Elements Suffer from Responsive Wonkiness](#)

[Width-Specificity in Page Layers](#)

[Exit Strategy](#)

[9 Extracting Your Way Out of Photoshop](#)

[Asset Extraction Is Like Pulling Teeth](#)

[Crop and Save](#)

[Copy Merged](#)

[Save for Web](#)

[Adobe Generator](#)

[Auto-magic Generation](#)

[Pixel Precision](#)

[Speaking Fluent Generator](#)

[Layer Naming as a Practice](#)

[Extract Assets](#)

[Setup](#)

[Extract](#)

[Setup](#)

[Downloading Assets via Libraries](#)

[Extracting Values](#)

[Generating CSS](#)

[10 Extending Photoshop](#)

[Building the “You” Version of Photoshop](#)

[Artwork](#)

[Subtle Patterns](#)

[Random User Generator](#)

[Social Kit](#)

[Pictura](#)
[Transform Each](#)
[DevRocket](#)
[Bjango Actions](#)
[WebZap](#)
[Composer](#)
[Layout Wrapper](#)
[RotateMe](#)

[Color](#)

[0to255](#)
[Adobe Color \(formerly Kuler\)](#)
[Adobe Color CC for iOS](#)
[Coolorus](#)

[Assets](#)

[iOS Hat](#)
[OtherIcons](#)
[Glifo](#)
[FlatIcon](#)
[TinyPNG](#)
[ImageOptim](#)

[Prototyping](#)

[Framer, Composite, and Stand In](#)
[InVision](#)

[Organization](#)

[GuideGuide](#)
[Renamy](#)
[Ink](#)
[psdiff](#)

[Miscellaneous Photoshoppery](#)

[ShortcutFoo](#)
[Photoshop Secrets](#)

[11 Remembering Etiquette](#)

[The Problem with Inheriting PSDs](#)

[What Is Photoshop Etiquette?](#)

[Improves Efficiency](#)
[Keeps You Organized](#)
[Creates Conventions](#)

Increased Importance in an RWD Workflow

Files

Name Files Appropriately

Store Assets Relative to PSD

File Accessibility

Layers

Name Layers and Be Accurate

Use Groups and Globalize Where Possible

Delete Unnecessary Layers

Images

Be Nondestructive

Use Blend Modes with Care

Be Aware of Resolution and Density

Type

Standardize Font Access

Don't Stretch Type

Control Your Text Boxes and Separate Them

Effects

Use Overlays Appropriately

Nail Tileable Images

Be Deliberate

QA

Proofread

Account for All Assets

Be Familiar with Browser Compatibility

12 Adopting a Completely New Workflow

Looking Back at Moving Forward

Full-Page Photoshop Comps Are Disharmonious with RWD

Designing in the Browser Helps, But Not As Much As We'd Like

2 Cups Browser, 1 Cup Photoshop

Vetting Direction Efficiently Is Critical

Style Can Be Established Through Small Exercises

Page-Building Is Easier with Component-Based Systems

Page Layers Makes Going from HTML to Photoshop Simple

[*New Extraction Tools Get Us Back to the Browser Quicker*](#)

[*We Can Customize Photoshop for RWD with Useful Third-Party Extensions*](#)

[*A Little Etiquette Goes a Long Way*](#)

[On Adoption](#)

[*Strategies for Getting Buy-in Internally*](#)

[*Strategies for External Getting Buy-In*](#)

[*What Happens When Things Go Wrong*](#)

[Adjusting Your Perspective on Tools](#)

[*Repurposing Tools May Be Better Than Getting New Ones*](#)

[**Index**](#)

1. Photoshop's New Groove

“We need to make this responsive.”

Whether you consider yourself a web design noob or pro, this declaration is as exciting as it is terrifying. Having some experience making responsive sites affords you only a small advantage over those who are just learning how. The way we design, even responsively, has evolved significantly in just two short years. That's the nature of the ever-changing Web, where the best tool you can have in your box is an eagerness to stay as up-to-date as possible.

That's why you picked up this book, isn't it?

At this moment, odds are you find yourself in a dilemma. *Responsive web design* (RWD) is becoming synonymous with simply web design, yet your process hasn't quite caught up. You've always used Photoshop, but its place in a responsive workflow seems a bit forced. *Do I really need to make that many comps? Are any of these newfangled design apps worth trying? There's got to be a better way!*

Even if you've done the responsive song and dance a few times, Photoshop most likely is no longer the workhorse of your web design workflow. Designing solely in the browser continues to pillage would-be Photoshop users, and new apps such as Reflow, Webflow, and Macaw present increasingly viable alternatives. The voices calling for the demise of Adobe's flagship app in web design are well-documented, and it's starting to make sense. You just might be the only one you know who still uses Photoshop for web design. Perhaps it's time to cut your losses and jump ship.

Yet there you are, still reading a book with “Photoshop” in the title. Stubborn. I like that about you.

Called Into Question

Photoshop is the one of the most polarizing topics among web designers today and has been ever since RWD came on the scene in 2011. I can remember the first time I really noticed it, back in September 2012. While scrolling through my Twitter stream I saw a link to a post on Treehouse's blog, titled “Responsive Web Design in the Browser Part 1: Kill Photoshop” (see [Figure 1.1](#)). Mind you, up until that point, I was already designing responsively but had no idea I wasn't supposed to be using Photoshop anymore.

Responsive Web Design in the Browser Part 1: Kill Photoshop

Josh Long • Design, Learn to Code, Make a Website • 16 Comments



Is this the end of photoshop?

If you're like me, then you've had a love/hate relationship with Photoshop for almost a decade. The software is bloated, it crashes right at the moment you forgot to hit save, and it produces a design that is nothing more than a useless and non-functioning image to send to clients. Despite loathing the software that is the staple of our industry, we can't live without it. High fidelity mockups have spoken in a way that we just can't beat with any other applications... until now. Let's explore the next stage in the evolution of the web and one where Photoshop is a waste of our valuable time. Here's how to kill Photoshop and do responsive web design in real-time with the only app we need: the browser.

Figure 1.1 The alternatives seem to be many, but the call to action from the design community seems to be clear: Kill Photoshop!

Now, I'm not easily dissuaded by one person's opinion, but the topic seemed to hit a fever pitch soon thereafter.

- ▶ "Has Responsive Web Design Killed Photoshop for Web Designers?" (www.boagworld.com/design/has-responsive-design-killed-photoshop-for-web-designers/)
- ▶ "Is Photoshop Dead?" (www.webdesignerdepot.com/2013/02/is-photoshop-dead/)
- ▶ "Photoshop Users: How to Switch to Sketch" (<http://blog.mengto.com/photoshop-users-how-to-switch-to-sketch>)
- ▶ "Khoi Vinh on Using Sketch Instead of Photoshop"

(www.creativebloq.com/khoi-vinh-using-sketch-instead-photoshop-6133901)

► “Photoshop Killer” (www.photoshopkiller.com)

The next thing I knew, I couldn’t find more than a handful of designer friends who said they still used Photoshop, or were willing to admit it anyway. They’d moved onto “more efficient” methods like using Sketch (www.bohemiancoding.com/sketch) or just designing with Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). It’s possible you may work somewhere that mandates you use Photoshop. But for designers like myself who have autonomy to use whatever tools and more or less processes they want, dumping a product I’ve used for my entire career wasn’t a trivial matter.

For the past few years, I’ve tried to find out as much as I can about why the switch from Photoshop is happening and who might still be using it. In my quest I’ve found a common concern shared by designers: Beyond it simply being a preference, Photoshop is intrinsically tied to their current employment. In other words, when some designers hear “They’re telling us to stop using Photoshop,” they internalize “I’m worried about my role on the team and potentially my career.” That seems pretty heavy for one tool, but don’t underestimate the impact it’s had on our industry.

Should you switch?

Stick in the Mud

Sure, you could ditch years of Photoshop familiarity, training, and cost. You could invest in a new web design app and never look back, at least until an even newer app comes out. Nobody would blame you, and they’d probably applaud you because you did it in the name of responsive web design efficiency. Everybody is doing it.

Still, I contend that jumping ship because of popularity isn’t good enough, not when the other option is simply trying harder to see the redeeming qualities of Photoshop and whether a better process is just a few tweaks away. I have a hunch you feel the same. That’s good because we’re about to take a deep dive into exploring how to get the most out of Photoshop without sacrificing workflow efficiency.

Fear of the Unknown?

I don’t think reluctance to drop Photoshop comes from being afraid to try new tools. I’ve tried a bunch of them. The appeal is there, and I encourage you to assess them for yourself. Fluid canvasses, adjustable breakpoints, and easy browser previews aren’t native to Photoshop. We can hope that these features will be baked in at some point, but that doesn’t seem likely. Adding them would require a major pivot from the iconic landscape of the product and likely frustrate users who are just fine with the current canvas.

However, the supposed “lack” of those features makes Photoshop a great environment for designing a single “moment.” Designing individual instances, in moderation, can help spur ideas to be introduced on a larger scale. Not everything about designing in a static medium is as bad as you might think. Granted, a lot of new tools have parallels to Photoshop in terms of drawing shapes and placing type. However, when handled appropriately, focusing on a single breakpoint can bring focus to a design. When I don’t have to be concerned about what shape navigation takes at a narrower or wider view, I can commit to exhausting an idea.

Can I Still Get by Without Knowing Code?

Equally important, this isn't an "anti-design in browser" cookbook for getting around knowing how to code either. Being able to design in the browser is essential when it comes to responsive web design, and this book has a whole chapter about it. I'm a staunch advocate of designers who can craft HTML & CSS because the structure it provides creates both possibilities and constraints for your design. The benefits aren't just individual, either. Ben Terrett of Gov.UK puts it this way:

"All of the designers can code or are learning to code. We call ourselves the design team because it's important to belong to a group with shared skills and experiences. This helps people develop their skills, support each other, and build a strong culture with shared standards."

—BEN TERRETT

(<https://gds.blog.gov.uk/2014/07/18/whats-the-design-process-at-gds>)

Don't worry, though. If you're a bit behind on your coding skills, there are wonderful resources to help you learn, like Treehouse (www.teamtreehouse.com), Code School (www.codeschool.com), and Codecademy (www.codecademy.com) to name a few. I won't be covering anything overly technical, but being able to bring your ideas to life in the browser has never been more critical (see [Figure 1.2](#)). (If not for lack of resources, what might be holding you up from learning some basic HTML & CSS?)

Learn HTML, CSS, iPhone apps & more

Learn how to build websites & apps, write code or start a business.

Free trial

How it Works

Learn to code interactively, for free.



Sign up

Login

Email

Username

Password

GET STARTED!

Or sign up with:



By signing up, you agree to our [Terms of service](#)

Learn By Doing

No setup. No hassle. Just learning.



Code School teaches web technologies in the comfort of your browser with video lessons, coding challenges, and screencasts.

[VIEW OUR COURSES](#)

[Or, learn where to get started](#)

LEARN: [ANGULARJS](#) [SASS](#) [RAILS](#) [JQUERY](#)

[OBJECTIVE-C](#)

Figure 1.2 The chasm between design and coding skills has plagued web design for years and still exists for many.

So if you're amenable to learning new tools and designing in the browser, what keeps pulling you back to Photoshop? For me, it's a tool I'm just too proficient with to abandon at the arrival of responsive web design. Proficiency is, and always will be, valuable to web design. And before you go disqualifying your own proficiency, note that I only recently got the Save for Web keyboard shortcut down (otherwise affectionately known as Save for Web Claw...just Google it). Proficiency isn't just about knowing your way around; it's about being able to ideate and render your design intent to the best of your ability. So, I'm declaring that it's OK to be a stick in the mud about this one, as long as you're open to figuring out how Photoshop's role is changing.

More Process Than Tool

Because there's no Convert to Responsive button in Adobe Photoshop (see [Figure 1.3](#)), we're going to have to talk about process a whole lot more than the new bells and whistles in Photoshop CC. (It would be pretty lame if there were such a button anyway.)

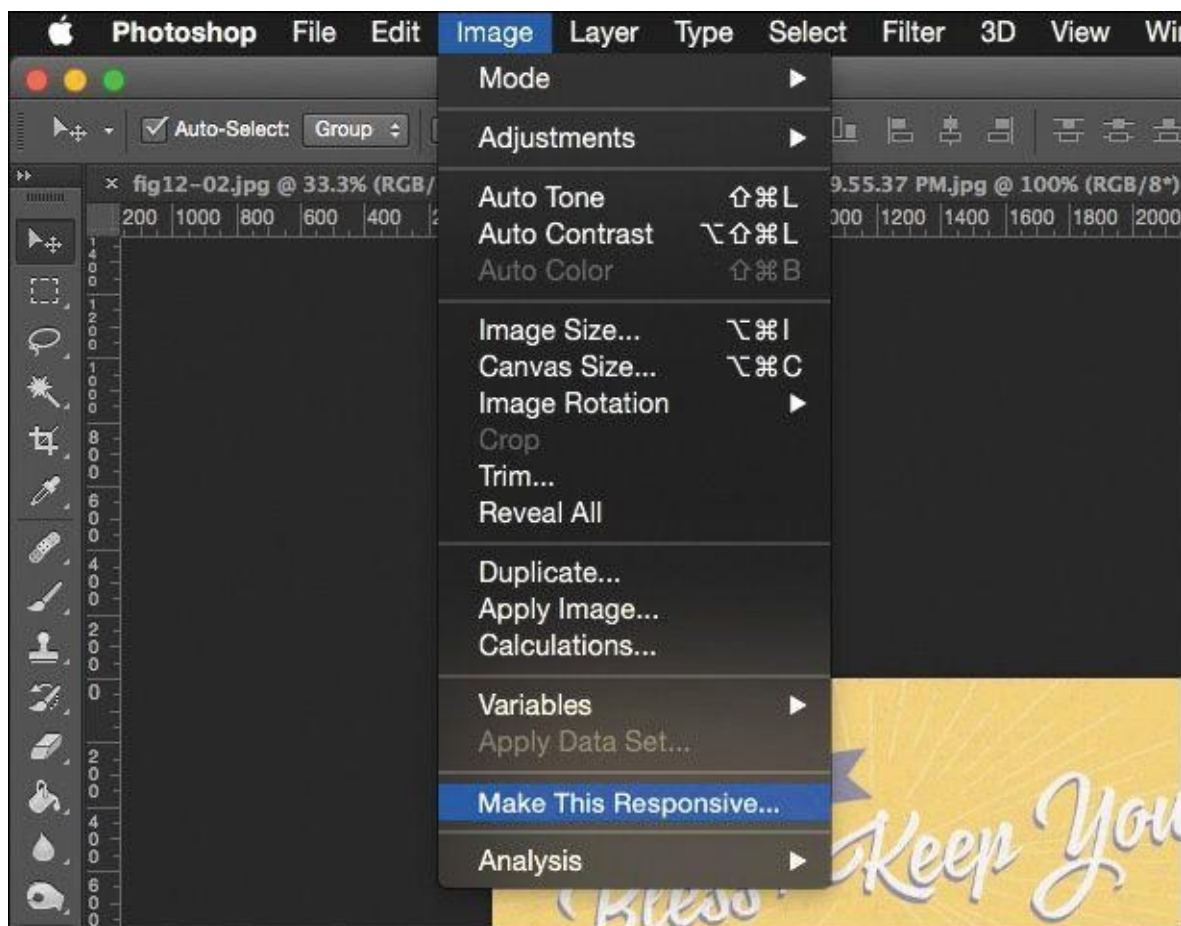


Figure 1.3 Have you found the magic button yet? It appears when you hold down every key except ~ on your keyboard. OK, I'm kidding...don't do that.

I've always maintained the most enjoyable part of web design is creative investigation. Designing for myriad screen widths provides ample opportunity for exploration, and while it often seems like a lot of work, it's work that shouldn't be automated. Each site design presents a set of challenges and problems to solve, and you can spot "templated" solutions from unique and well-considered ones. Design thinking and design tools need to work in

beautiful choreography. To make your responsive workflows successful, you need to shift your focus from “Which tool should I use?” to “Where does it make sense to use each tool in my process?”

Today’s workflows are about staying nimble. I’m confident you can stay nimble using Photoshop, but you may need to rework your traditional approaches to do so.

A Battle of Two Short Words

I almost made a horrible mistake naming this book.

I was *this* close to titling it *Responsive Web Design in Adobe Photoshop*. You just checked the cover to see what the difference is, didn’t you? But the distinction may not be clear or significant to you at the moment.

What’s the difference between *Responsive Web Design in Adobe Photoshop* and *Responsive Web Design with Adobe Photoshop*?

Distinguishing *with* from *in* is vital. For years, designers have used Photoshop as a program to do web design “in,” in its entirety. The “Photoshop phase” of a project was half of the work. Once our infamous “Home.psd, Landing.psd, Interior.psd” mock-ups were approved, we’d wipe our hands clean and move on. Months of project timelines were spent assessing work done in Photoshop.

That was when we were designing for a single screen, and even then you could argue how inefficient and inaccurate of an approach it was.

Designing for multiple screens places greater emphasis on staying nimble. You’ve probably already experienced how backlogged a project gets when you’re making three templates at three sizes each. How about more than ten templates at more than five different sizes? Overwhelming. No doubt you’re under the gun to shorten the Photoshop phase, or perhaps you’re just genuinely interested in a more efficient method. Logically, if you’re cutting down design time in Photoshop, you have to increase the time you spend in HTML & CSS. The trick, as you’ll discover, is continuing to work “with” Photoshop even after you (or a developer) start toying with code.

Responsive web design isn’t something you can do inside the silo of Photoshop. You can, however, use Photoshop alongside the browser (and a bevy of other tools) to help you design responsively.

Not on the Menu Tonight

Now is as good a time as any to discuss some of the topics I won’t be covering in depth, don’t you think?

The Core Tenets of Responsive Web Design

On the RWD side, I’m assuming you’re familiar with its core tenets: media queries, flexible grids, and fluid images and video (see [Figure 1.4](#)). Therefore, we won’t be spending time defining what’s already been so brilliantly defined by the inimitable Ethan Marcotte (www.alistapart.com/article/responsive-web-design). He also published the penultimate book on the topic, *Responsive Web Design* (www.abookapart.com/products/responsive-web-design). If you haven’t read it, do so on

my recommendation and that of every other person who owns it.

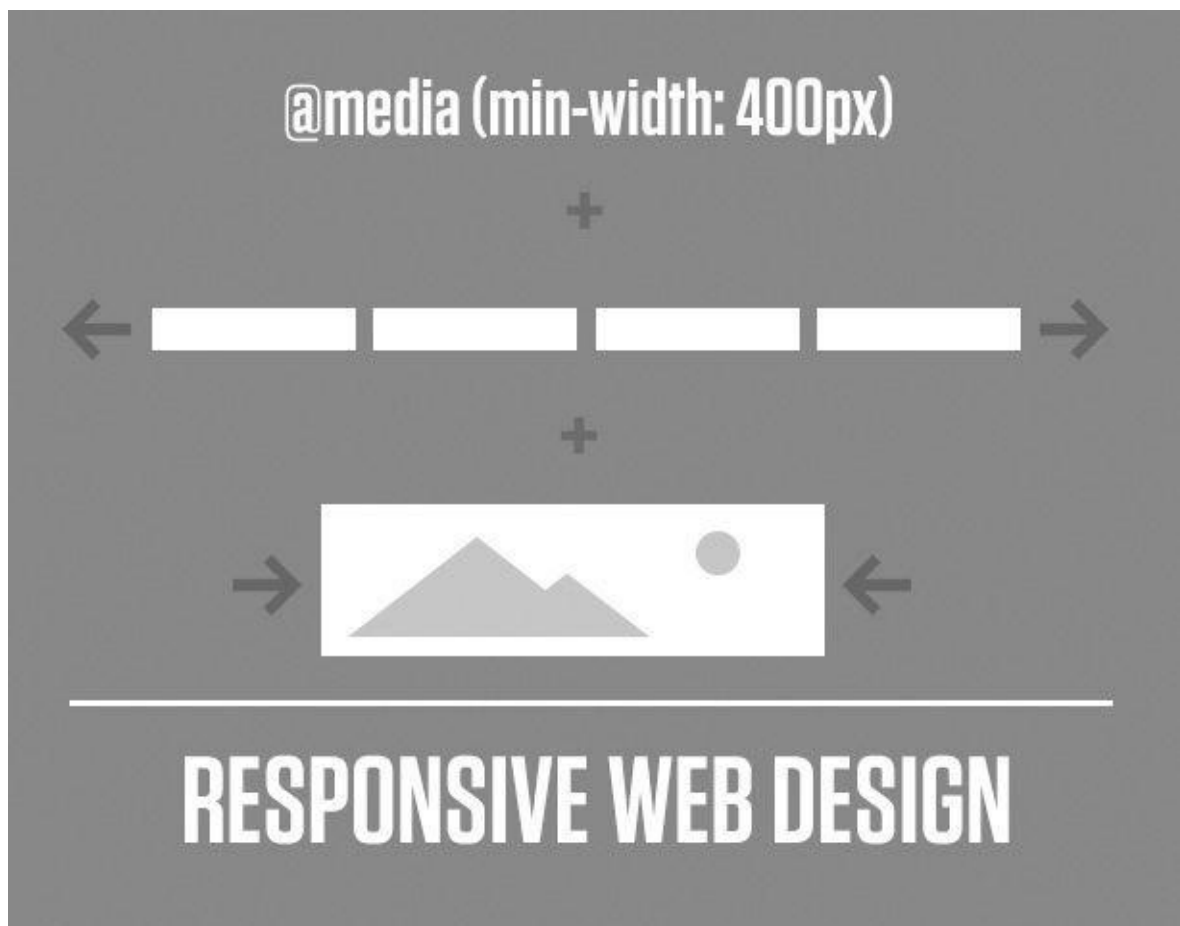


Figure 1.4 RWD = Media Queries + Flexible Grids + Fluid Images. Got it? Good.

Responsive Patterns

There are a ton of responsive patterns and conventions that the fine people in our industry continue to develop, and it would be nearly impossible to detail them in the scope of this adventure. For example, the number of patterns available for navigation at different screen sizes spans from stacking items vertically to triggering an off-canvas menu like Facebook of yesteryear. I'll touch on a little of that, but a much more comprehensive resource I often use is *This is Responsive* (<http://bradfrost.github.io/this-is-responsive>), curated by one of the finest people on the Web (and IRL), Brad Frost.

Performance

Lastly, there's a critical component to responsive web design that we designers tend to overlook at our own peril: performance. Some of the most gorgeous sites not only take forever to load on a standard mobile connection but also end up sending your data plan overage fees through the roof. Why is this? Among other things, uncompressed images are to blame, but there are a ton of other factors for slow performance that are a bit too technical for our discussion. Scott Jehl's *Responsible Responsive Design* (www.abookapart.com/products/responsible-responsive-design) tackles the impact of the choices we make.

Photoshop Basics

On the Photoshop side, I'll assume you're familiar with the interface. You've comped