

Mobilizing Web Sites

DEVELOP AND DESIGN

Strategies for Mobile Web Implementation



Kristofer Layon

Mobilizing Web Sites

DEVELOP AND DESIGN

Kristofer Layon



Mobilizing Web Sites: Develop and Design

Kristofer Layon

Peachpit Press
1249 Eighth Street
Berkeley, CA 94710
510/524-2178
510/524-2221 (fax)
Find us on the Web at: www.peachpit.com
To report errors, please send a note to errata@peachpit.com
Peachpit Press is a division of Pearson Education

Copyright © 2012 by Kristofer Layon

Project Editor: Michael J. Nolan
Development Editor: Jeff Riley/Box Twelve Communications
Production Editor: Myrna Vladic
Copyeditor: Deborah Burns
Technical Editors: Zachary Johnson and Mike Resman
Proofreader: Jan Seymour
Interior Design: Mimi Heft
Cover Designer: Aren Howell Straiger
Cover Production: Jaime Brenner
Compositor: David Van Ness
Indexer: Valerie Haynes-Perry

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 nor Peachpit 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

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.

ISBN 13: 978-0-321-79381-2

ISBN 10: 0-321-79381-1

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

For Katie.

*I've written two books,
but I still can't put into words
how much I love you.*

ACKNOWLEDGMENTS

The goal of this book is twofold: to ease your entry into mobile design and development with examples and case studies, and to encourage and inspire you to start doing mobile work today, iteratively and in baby steps, by mobilizing web sites that you already work on.

So this book is not on the bleeding edge of mobile design approaches and techniques. Rather, it was inspired by many other designers, writers, and speakers who have informed my approaches to mobile design, content strategy, usability, and product management over the past several years. To document and pay homage to these sources of inspiration and best practices, I'm grateful that a number of them were willing to coordinate and contribute ideas and interviews about their work in mobile design and development: Ethan Marcotte, Colleen Jones, Luke Wroblewski, and the following people at Happy Cog—Rawle Anders, Stephen Caver, Greg Hoy, Mark Huot, Jenn Lukas, Yesenia Perez-Cruz, Greg Storey, Russ Unger, and Jeffrey Zeldman.

I'm also grateful for the contributions of Zachary Johnson, who was my main technical editor again after having that role for my first book. His corrections and suggestions for the HTML, CSS, and JavaScript examples in this book did much to improve and clarify my work. And I am grateful for the additional technical work of Mike Resman, my brilliant intern at the University of Minnesota who collaborated on the development of the first mobile web project I ever worked on. It's his technical skill that made that first project succeed, and it's also his work that made Chapter 10 of this book possible, as he was either the writer or editor of all of the PHP code that is featured there.

And thanks to Michael Nolan at Peachpit who had enough faith to have me write a second book. I think Michael and I were made for each other, as he feels that signing a project isn't necessarily about finding the right *expert* for a given topic. Rather, it's about finding the right *person*, at the right *time*, who has the right *idea*, and who can execute that idea *clearly* and *on time*. I will certainly agree that I am not a mobile expert, but I was excited to propose a book that sought to bring mobile design within reach of the everyday web designer and developer. I firmly believe that acceptable mobile design does not require an expert nor a ton of time and money—it just requires the decision to give it a try, and a commitment to keep improving on what you learn. That's what I am doing, and if this book inspires other people to get into mobile design and development, I will consider it to be a success.

Thanks also to my development editor Jeff Riley and the editorial and design staff at Peachpit, whose work led to this book looking as good as it does. And thanks to my supervisor at Capella Education Company, Jason Scherschligt, who was willing to hire me last spring as Capella's new Mobile Product Manager. I have the best mobile job imaginable and work with some of the brightest mobile designers and developers there are.

Finally and most importantly, I thank my family: Katie, Sarah, Grace, Emma, and Anne. You tolerated me leaving for work an hour earlier every day for several months to write this book. It wouldn't have happened without your support and encouragement.

CONTENTS

	Foreword	ix
	Introduction	x
CHAPTER 1	MOBILIZING USER EXPERIENCE DESIGN	1
	Devices aren't Mobile: People Are	2
	Mobile = Doing	3
	Mobile = Busy	4
	Mobile = Distracted	6
	Mobile = Relaxed	8
	Mobile = Tasks	10
	Talk to Your Customers	11
	Write User Stories	12
	Break User Stories Down into Design Requirements	14
	Create a Prioritized Story Backlog	15
	Wrapping Up	19
CHAPTER 2	MOBILIZING LAYOUT	20
	Standards-Based Layouts are Already Responsive	22
	What Makes It Fixed-Width?	23
	Un-Fixing the Width	24
	Shuffling the Squares	25
	And Now Introducing Our Web Site	28
	Wrapping Up	37
	PROJECT INTERVIEW 1:	
	ARIZONA STATE UNIVERSITY	
	AND THE MTV O MUSIC AWARDS	38
CHAPTER 3	MOBILIZING NAVIGATION	46
	Watches	48
	Car Interfaces	50
	iOS App Interfaces	52
	And Now... Mobilizing Navigation	54
	Wrapping Up	59

CHAPTER 4	MOBILIZING IMAGES	60
	A Holistic Approach to Mobile Images	62
	Inline Images	63
	Responsive Photo Grids	68
	Wrapping Up	72
CHAPTER 5	MOBILIZING TEXT	74
	Take a User-Centered Approach to Mobilizing Text	76
	By Which Measure: A Point or an em?	77
	Using Show and Hide	82
	Wrapping Up	89
	PROJECT INTERVIEW 2:	
	THE BOSTON GLOBE	90
CHAPTER 6	MOBILIZING FORMS	98
	Mobile Can Improve the User Experience of Web Forms	100
	Input Types	101
	Wrapping Up	109
CHAPTER 7	MOBILIZING SOCIAL MEDIA	110
	Mobile “Comes Free” with Most Social Media	112
	Twitter	113
	Facebook	114
	Google+	115
	Quora	116
	LinkedIn	117
	YouTube	118
	Connecting to Social	119
	Wrapping Up	123
CHAPTER 8	MOBILIZING CONTENT STRATEGY	124
	Understanding Mobile Context First Leads to a Better Content Strategy	126
	Mobilized Design Prepares You for Usability Evaluation	127

	Usability Evaluation is the First Step in a Good Content Strategy	128
	Content Strategy Prepares You for Mobile First Design	130
	Wrapping Up	131
	PROJECT INTERVIEW 3: INTERCONTINENTAL HOTEL GROUP	132
CHAPTER 9	OTHER WAYS TO MOBILIZE: JQUERY MOBILE	140
	The jQuery Mobile Web Framework	142
	Advantages and Disadvantages of jQuery Mobile	143
	Head	144
	Body and Header Bar	146
	Navigation	147
	Tool Bar and Buttons	148
	Modal Dialog Menu	150
	Forms	152
	Landscape	153
	Wrapping Up	153
CHAPTER 10	OTHER WAYS TO MOBILIZE: DEVICE DETECTION AND PHP INCLUDES	154
	What is PHP?	156
	Using PHP	157
	Wrapping Up	171
	Conclusion	171
	Index	173

FOREWORD

Baby steps.

The phrase gets bandied about so much these days that we often lose sight of its significance.

A baby's first steps aren't just a few simple motions that get them started. They're a profound transformation from seeing the world move around us to taking part. Baby steps change our entire perspective and open up countless new opportunities to not only walk but run, skip, and dance. They're our entry into a whole new way of living.

The book in your hands (or on your screen) is also an entry into something new. Something so transformational it's fundamentally changing how we interact with our knowledge, our places, and with each other. That something is how we experience the Internet on mobile devices.

Always with us, always connected, fast and powerful mobile devices are taking over and redefining how we find answers, make purchases, share with others, and get things done. According to Jeremy Weinstein (<http://goo.gl/Noq9L>), in 2009, 50% of all new Internet connections worldwide came from mobiles. Gartner reports that by 2013, mobiles will overtake PCs as the most common web access devices worldwide (<http://goo.gl/J64Zn>).

If you've got an existing web site, chances are you're already noticing the transition and seeing your mobile traffic rising fast. But if do have an existing web site, do you need to start over to be relevant in the mobile age? How can you and your site adapt to such rapid change?

Baby steps.

Kris's book is filled with them—little but powerful things you can do today to adapt your current site to the changing web and the growing importance of mobile devices. These techniques will get you started without having to completely rebuild and redesign your current web site. But like real baby steps, they'll also open your eyes to new opportunities and ways of seeing the web. From there, who knows how far you'll go.

So what are you waiting for? Take those first steps now.

—Luke Wroblewski
Principal, LukeW Ideation + Design
Author, *Mobile First* (A Book Apart)
November 11, 2011

INTRODUCTION



FIGURE I.1 The IDS Tower in Minneapolis, the tallest office building downtown when it opened in 1974.

Whether it's "mobile first" or "mobile later," we are in the age of mobile computing and web browsing.

Don't let your web sites get left behind. Mobile later is better than not going mobile at all! More importantly, it can be your first step toward mastering mobile web design.

OVERCOMING THE FEAR OF "MOBILE NEVER"

Over the past few years, I have had the privilege of attending some great conference presentations about mobile app design and the mobile web. And if you have an interest in mobile and are reading this right now, you have also probably learned some valuable lessons and techniques to help you design new web sites for presentation on mobile devices.

The trouble for many of us, however, is that we often just go back to work and keep dreaming about designing that first mobile web site ... as we continue to work on our existing desktop-oriented projects. Many of us are in-house designers who help maintain and update several existing web sites, or maybe just one really large one. Or, we have contracts to continue maintaining and updating existing web sites for our clients.

But in these cases, we're not often in a position to propose an entirely new site design so that we can achieve our mobile goals.

So the rallying cry of "mobile first" might get us excited and inspired, but where does it leave those of us who are faced with "mobile later?"

Worse yet: what if we feel like our circumstances are keeping us locked into "mobile never?"

THIS IS NOT A UNIQUE PROBLEM

Among other factors, the allure of creating something new is what drives many of us into the field of web design. Images, text, and code: the thrill of filling an empty browser window with a new interactive design that we have created is exciting and satisfying. So when we are instead faced with incremental updates or improvements to existing web properties, it can feel more mundane. Where are the creative opportunities in that?

But sometimes we forget that working on the web is not any different from working with other media, materials, or mechanical systems. People get into a variety of creative fields because of the allure of starting from scratch and filling

a blank canvas with something stunning and new—yet this is often the exception and not the rule. Take buildings as an example.

People who grow up wanting to design buildings often end up going to architecture school and, when they complete their programs, graduate as architects. Similarly, people who prefer a more hands-on experience of craftsmanship and construction attend a trade school to learn how to become a carpenter, mason, or other tradesperson. Regardless of the role, people who want to be engaged in the building design and construction fields are just as eager as web designers to make their mark on the world. They want to build something exciting that they can be proud of. And they usually dream about building something new (**Figure I.1**).

But for every big new construction project, there are many more renovation or improvement projects. These projects are what keep most architects and construction tradespeople busy most of the time.

I am fortunate to have worked in the architecture field prior to becoming a web designer, so this comparison is more than just metaphorical for me. As a graphic designer with a background in architectural history, I spent several years working for architects on a variety of tasks including historic building inventories and the design of client presentations. Despite my passion for new architecture (at that time, my plan was to get into architecture school after gaining some experience in the field while my wife was in graduate school), most of these firms' projects were renovations, additions, and reuse projects (**Figure I.2**).

But let's face it: often what is old becomes new again anyway, right? A renovation or reuse project isn't really just an old project with a new layer. It is a new project with a different set of constraints, and a unique set of existing conditions and context.

MOBILE LATER

This is a cue to the world of web designers and developers: we, too, got into our field because we were excited to make something new. But as I noted earlier, this isn't always the case.

Or is it?

Re-contextualizing our work using the architecture example, we can see that a new web site does not always need to start from scratch. And in this particular case, a mobile web experience also does not need to start with a new web site. "Mobile later" can indeed work; we do not need to wait for "mobile first." And, actually, a

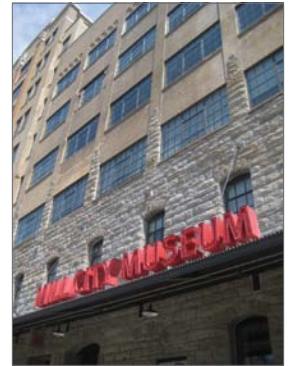


FIGURE I.2 The Mill City Museum in Minneapolis, which opened in 2003 as a museum, office, and apartment building. It was originally built in the 1870s as a flour milling complex.

great first step toward that elusive new “mobile first” site design is to start mastering the mobile web by taking baby steps and applying mobile techniques to an existing site design that you’re already familiar with.

So how does this work, and where does one begin? That existing web site that you would like to mobilize for action in the 21st century of on-the-go web browsing seems rather hunkered down in a fixed-width, grid-based design that appears to be hopelessly mired in the 20th century.

If this is your fear, this book is written for you.

ABOUT THIS BOOK

As soon as you read that sentence, you may have thought, “It is? *Why* is this book for me? And why should I buy another book about mobile design? Hasn’t this been written already?”

Yes and no. I have been collecting and reading a number of great books and articles about mobile design for the past few years, and they are all good in their respective ways. But they tend to fall into one of these categories:

1. Research-based analysis about the growth of mobile device use, and why now is the time for everyone to design for mobile devices. *Today*.
2. Higher-level, conceptual thinking about what makes a good mobile app or web site and how to plan such a project.
3. Visual examples of how to design screen layouts and user interfaces.
4. Technical, code-based techniques for developing web pages in a more responsive manner (usually focusing on CSS methods).

And when used as a library, these books and articles are all fabulous. In fact, I will refer to some of these writings in this book and its companion web site (www.mobilizingwebsites.net): they helped me get to where I am today, serve as a foundation for my ongoing work and writing, and are part of a shared base of knowledge for our evolving field.

But here are some things that distinguish this book:

1. As the title of the series implies, this book bridges the worlds of design and development. It gets you thinking about mobile design in a very holistic

manner, from the earliest stages of ideation and discovery at the very beginning of a mobile web project to usability evaluation at the end.

2. This book applies mobile planning, design, and coding techniques specifically to web sites that were initially designed for desktop display, which is what many of us deal with on a day-to-day basis. So all of the examples shown in the book including available code samples and pages to view at the book's companion site, <http://www.mobilizingwebsites.net>, are based on the common set of constraints of an existing fixed-width, grid-based web site.
3. The book organizes its suggestions and examples according to distinct mobile web challenges: how to present layouts, navigation, images, and text on small screen devices.
4. More importantly, this book helps you think about web mobilization *incrementally*. Mobilizing user experience, interface design, and content strategy does not need to be an all-or-nothing endeavor. Think about improving your web site like you would think about improving software or any product that is complex and long-lasting (and again, think about buildings). Smaller, incremental, but ongoing improvements can be just as challenging and rewarding to define and tackle as all of your improvements at once.
5. Finally, this book makes the case that after you become familiar with mobile constraints, techniques, and opportunities, and apply them to an existing site, you are then positioned to learn and do much more. Then you can have your site evaluated by users, rethink your content for mobile, and make the most informed decisions you can about making continued improvements. At that point, a more comprehensive "mobile first" redesign won't seem so daunting: it will feel like the right thing to do.

So if this is the book that mobilizes you into action today, and helps you apply mobile ideas and techniques that inspired you before you left them at the threshold of your office due to the common constraints of dealing with legacy web sites, I high-five you from afar.

Go forth and mobilize!

4

MOBILIZING **IMAGES**



Images have been making our communications and storytelling come alive since cave drawing, oil painting, and photography. What are some considerations and techniques for images in mobile web presentation?

A HOLISTIC APPROACH TO MOBILE IMAGES

With digital photography and illustration, especially on the web, we have been increasingly accustomed to our CSS-empowered browsers gracefully processing image placement and scaling. And now with CSS3, the options keep growing to include fancy border treatments, more sophisticated layout options, and more.

So in modern browsers, we might expect the transition from desktop to mobile to be similarly magical and trouble-free. And, in many respects, it can go pretty darn well. But mobilizing images is about more than the technical details and relying on the browser to do everything. Let's examine some design considerations, along with their technical solutions, so that you can approach mobilizing images in a holistic manner.

When it comes to mobile images, you might initially jump to the conclusion that images should always be smaller because the screens are smaller, so therefore the images should take up less of that precious real estate.

Which can be true in some cases. But not always!

SOMETIMES MOBILIZING MEANS SMALLER ...

As Figure 3.7 illustrated at the end of the previous chapter, one of the issues you can run into is an image that was originally sized to display at its full size. Consider the blue and white Finnish flag at the top of the screen as an example of this. It is 150 pixels wide and 90 pixels tall. For the original 950 pixel wide design of the desktop experience, this flag was sized in Photoshop and intended to be displayed at 100%.

So the CSS for this image in `main.css` is:

```
img#flag {  
    margin-right: 15px;  
    border: 1px solid #999;  
}
```

Granted, there are two issues going on in this area of the design at the moment: a flag image that may be too large, and the “Design in Finland” text that is also too large for the space of the small screen. We will deal just with the image issue here.

The solution for the image, as you have probably guessed, is to size it down a bit. Fortunately, our HTML allows us to do this rather conveniently:

```

```

Note that there are no width or height attributes in the `img` tag. And this is exactly what we want when we are relying on CSS to do the heavy lifting of styling and positioning—keep the HTML light and airy, providing just a touch of structure to our content.

So note that if you have `img` tags with width and height attributes, and you’re striving for a responsive, mobilized design, you will want to strip out those attributes and have your CSS do the sizing; set an `id` or `class` for each image as you do this.

FIGURE 4.1 Our resized Finnish flag, no longer dominating the header with its full glory of 150 pixel width.



Getting back to resizing the image, adding a new CSS rule in `mobile.css` for this image looks like this:

```
img#flag {  
    width: 25%;  
    margin-top: 20px;  
}
```

The adjustments include narrowing the width to 25% which, on a mobile screen of 320 pixels, results in a more modest flag that is 38 pixels wide instead of 150. I also added a margin of 20px above the flag. Previously, it was flush to the top of the page and looked a bit goofy in the mobile presentation.

But other than that, I'm taking advantage of the cascading nature of CSS and not redefining the border or right margin of the flag in `main.css`. Consequently, those properties are still applied to the flag and the result is a flag with a narrow gray border and some right margin to buffer the headline text, just as before (**Figure 4.1**).

Perfect!

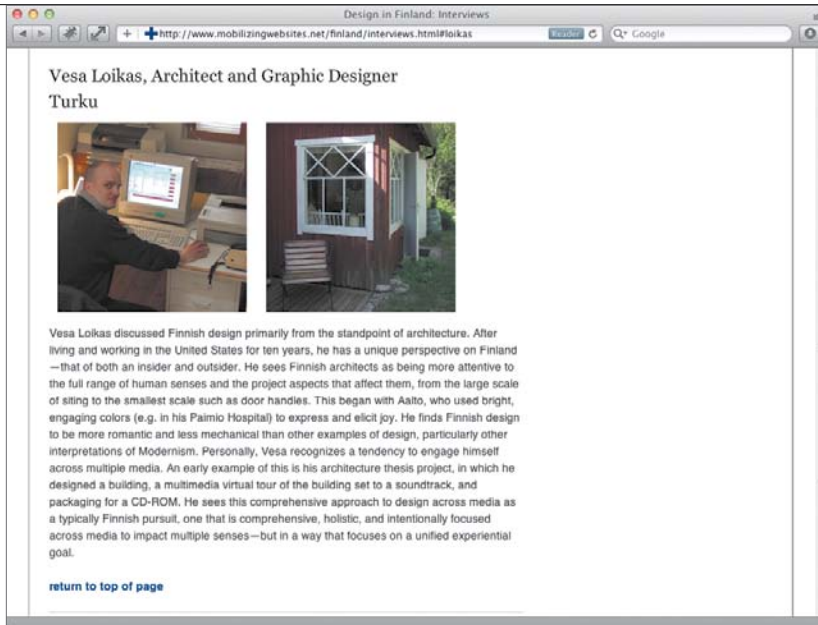


FIGURE 4.2 The interview page when viewed in the desktop version of Safari.

... AND SOMETIMES MOBILIZING MEANS LARGER

In other cases, you may not want to use your images at 100% or even size them smaller. There are cases where you will want to size them larger instead.

Seriously? Make them bigger when you have less screen to work with?

Yes. Seriously.

To see why, let's take a look at the images in the interview section of the Finnish design web site (**Figure 4.2**). In a desktop environment, they have been sized to be adjacent to each other. This layout is attractive and works quite well, and the image size (40% of their 600 pixel width) is just right for showing detail.

FIGURE 4.3 The interview page when viewed in the mobile version of Safari.



But when viewing this page on a mobile screen (**Figure 4.3**), I feel that the images are kind of small. The layout itself is still fine, but if I want to emphasize the images as content, I'm not certain that it's particularly successful here.

So the lesson here isn't nearly as technical as it is contextual and editorial: if you are assessing the success of images in mobile presentation, don't hesitate to make them *larger* to make them work better on *smaller* screens.

In this case, the adjustment to make in the CSS is fairly simple, going from this in `main.css`:

```
.interviewphotos img {  
    width: 40%;  
    padding: 10px;  
}
```

to these additional rules in `mobile.css`:

```
.interviewphotos {  
    width: 100%;  
}  
.interviewphotos img {  
    width: 90%;  
}
```

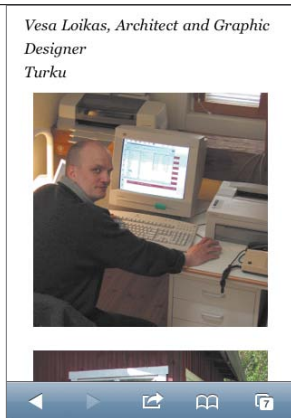


FIGURE 4.4 (left) The interview page when viewed in the mobile version of Safari.



FIGURE 4.5 (right) As you scroll the interview page in mobile Safari, the larger images are still manageable. There's more scrolling, to be sure—but it is still reasonable.

These changes do two essential things:

1. The change to the `interviewphotos` class makes the div's width 100% of the screen width.
2. The change to any images contained in an `interviewphotos` div results in them being 90% of the div width (or 90% of the screen width).

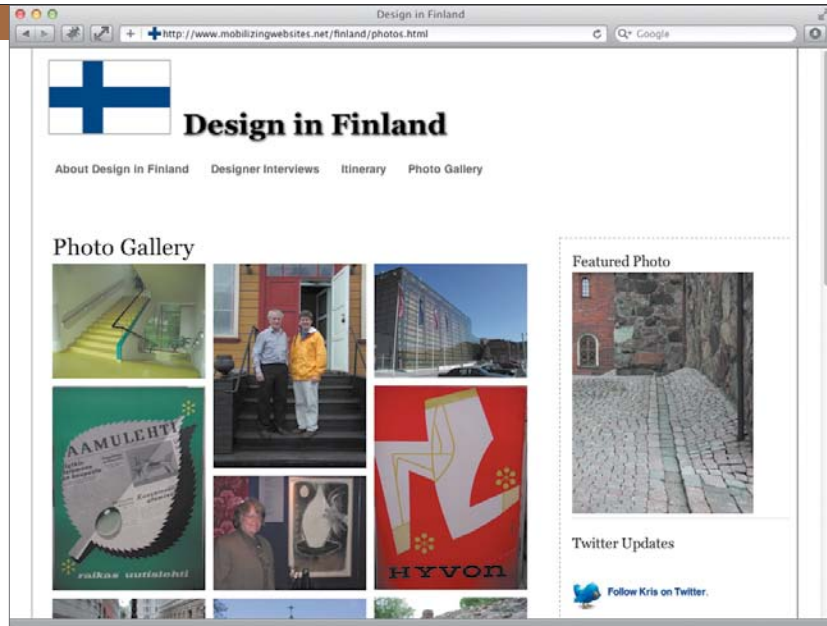
The redefinition of `interviewphotos` and `interviewphotos img` results in this mobile presentation of the images as shown in **Figure 4.4**.

Granted, the photos stack now and increase scrolling. Might too much of this be a problem on a mobile device? It certainly could—I'm not suggesting that stacking everything works in every case, so you have to make your own judgment calls on a project-by-project (and even screen-by-screen) basis. But the scrolling in this example is not outrageous by any means, and as you go from image to image, the text beneath the images resumes and keeps prodding the user to scroll. It's a natural flow that is aesthetically and contextually pleasing (**Figure 4.5**).

So we have examined two ways to handle inline images. Deciding when to make something smaller or when to make something larger is a decision to make based on the amount and purpose of your image content.

RESPONSIVE PHOTO GRIDS

FIGURE 4.6 A photo gallery using CSS3 columns. Living in the future!



There are a variety of techniques for integrating photos and other images into web sites, and among them are some cool techniques for making larger image layouts responsive.

The responsive photo grid used in Design for Finland's Photo Gallery uses two fantastic CSS3 properties, `column-count` and `column-gap`. These are properties that have support in some of the most popular browsers (Firefox, Safari, Opera) and, notably, in their mobile offspring.

So first, let's take a look at what this photo grid looks like in a regular desktop browser. The idea is to tile photos and have them flow through a series of columns from left to right across the page. The approach is based on how newspaper and magazine columns work in print: when a column fills, the flow moves over one column to the right and automatically starts filling the next column. **Figure 4.6** shows the result with 20 photos.

CHRIS COYIER'S TECHNIQUE

One of the techniques that I particularly like is demonstrated by the fine web craftsman Chris Coyier (<http://chriscoyier.net>). To learn more about how this is implemented, visit <http://css-tricks.com/13372-seamless-responsive-photo-grid/>.

So yeah, it looks suspiciously like a table doing the work here, doesn't it? Well don't you dare accuse me of something so gauche! There's just the tiniest of clues in the HTML that points to what is going on:

```
<div id="grid">






...
</div>
```

That's right, other than a series of `img` tags, there is nothing to the HTML markup at all—just the `div` id.

So what does the CSS look like? Well, like this in the site's `main.css` file:

```
#grid {
  line-height: 0;
  -webkit-column-count: 3;
  -webkit-column-gap: 10px;
  -moz-column-count: 3;
  -moz-column-gap: 10px;
  column-count: 3;
  column-gap: 10px;
}
```



```
#grid img {  
    width: 100% !important;  
    height: auto !important;  
    margin-bottom: 10px;  
}
```

Wow, that's it? Indeed—sometimes this great CSS3 stuff is as magical as it looks! All you need to do is use the three flavors of `column-count` as one needs to with any CSS3 property these days, to cover yourself for both WebKit and Mozilla browsers, as well as to future-proof your CSS when browser prefixes are no longer necessary.

Then, you just define a spacer using `column-gap` if you do not want the images to be flush against each other. This takes care of the behavior for the div that wraps the images, `#grid`. Then for any image located inside the div, the width is specified to be 100%, the height auto, and each image has a 10 pixel margin below it to match the 10 pixel gap in between the columns.

How does this respond to mobile browsers? Here's a wonderful lesson to savor from this use of CSS3: the easier it is to define a behavior in CSS, the easier it is to modify it for mobile responsiveness.

So hold on to your hat, because all you need to do is throw in a few media queries to have this layout respond to smaller screens. For the case study web site, I have chosen two different responses, one for screens 320px or smaller and another for screens up to 480px. Here's what you add to `mobile.css`:

```
@media (max-width: 480px) {  
    #grid {  
        -moz-column-count: 2;  
        -webkit-column-count: 2;  
        column-count: 2;  
    }  
}
```

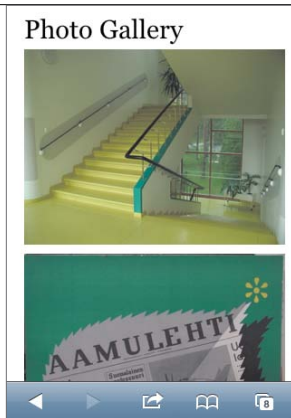


FIGURE 4.7 The responsive photo grid in portrait orientation as viewed in Mobile Safari.



FIGURE 4.8 The same responsive photo grid in landscape orientation.

```
@media (max-width: 320px) {
  #grid {
    -moz-column-count: 1;
    -webkit-column-count: 1;
    column-count: 1;
  }
}
```

Adding this to the site's `mobile.css` file does some quick and dirty magic, reducing the photo grid to two columns at 480 pixels or smaller (the size of the iPhone screen in landscape orientation) and to a single column at 320 pixels or smaller (and if you want to be more specific about targeting more specific screen sizes, adjust your media query accordingly).

So let's take a look at the mobile versions of this flexible photo grid in **Figures 4.7** and **4.8**.

I just love the simplicity of this responsive grid, and I feel like I could experiment for hours optimizing layouts for specific ranges of screen sizes (but I'll let you take it further if you wish!). And I am also a huge fan of the asymmetrical layouts that can result by interspersing landscape- and portrait-oriented images in this layout—I think the results can be really beautiful.

So continue to have fun with image layouts in your web sites, and do not feel constrained by mobile presentations of your images. A design constraint is just another name for a design opportunity!

WRAPPING UP

Optimizing the presentation of images on mobile devices can be done with the right application of CSS properties. Sometimes merely resorting to tried and true CSS2 properties in the case of resizing inline images or their container divs will do the trick. And in the case of using a flexible image grid, the extra magic that CSS3 can leverage makes the work (and the results) even more fun and responsive.

But do not let the two examples in this chapter suggest that these are the only directions to take. This case study is meant to be more than instructional—I hope it's also inspirational. Please take mobilization further in your own designs. Experiment with CSS in ways that allow your specific web site to respond to mobile conditions in the ways that it was meant to, way back when it was first designed!

This page intentionally left blank

INDEX

NUMBERS

768 pixels, using for screen width, 31

SYMBOLS

character, using in PHP, 166
/* comment */, using in PHP, 166

A

Android, Reason for Contact menu, 107
Arizona State University project interview, 41–45

B

Berners-Lee, Tim, 156
Blueprint framework, 23
 redefining widths of divs, 32–33
 rescaling classes and IDs, 32–33
body content
 paragraphs, 78–80
 subheadings, 80–81
borders, removing, 33
Boston Globe project interview, 92–97
 image techniques, 97
 scale, 92–97
 story presentation, 96
Bringham, Robert, 79
browser width, modifying, 30
browser windows. *See also* mobile browsers; web browsers
 resizing, 26
 responsiveness of viewports, 27

C

Calculator iOS app, 52
car interfaces
 complexity, 51
 position, 51
 quantity, 51
 size, 51
.com key, holding in iOS, 105
contact form, 101. *See also* forms
 “email” input type, 103
 First Name field, 102
 HTML5 input type, 104
 “tel” input type, 106
 “text” input type, 102
 “URL” input type, 105

content, displaying divs of, 35–36
content management system, using, 137
content strategy
 Intercontinental Hotel Group project interview, 135–138
 mobile first design, 130–131
 usability evaluation, 127–129
Coyier, Chris, 69
CSS framework
 fixed-width, 23
 un-fixing width, 24
CSS layout, fixed-grid, 27. *See also* mobile.css stylesheet
CSS navigation, 55–56
CSS3 columns, using in photo gallery, 68
customer satisfaction, levels of, 16–18
customers
 mobilizing, 11
 product performance requirements for, 17
 talking to, 11

D

Design in Finland web site. *See also* web sites
 About Design in Finland screen, 147–148
 browser width, 30
 contact form, 101
 fixed width, 28
 header and footer, 29
 heading, 77
 horizontal navigation, 29
 introductory paragraphs, 79
 jQuery Mobile’s “Swatch B,” 146
 linking to mobile stylesheet, 30–31
 media query, 31
 mobile CSS, 32–35
 mobile HTML, 30–31
 in Mobile Safari on iPhone, 29, 32, 34
 modal dialog menu in jQuery, 150
 multi-column layout, 29
 non-mobilized version, 29
 redefining width, 32
 resetting div widths, 54
 resized Finnish flag, 64
 resizing to 16pt, 78
 in Safari on Mac laptop, 28
 screen width, 31
 set in Georgia typeface, 170
 social links, 122
design specifications, writing, 12–13
device detection script, 165–166
 head_inc.php file, 166
 is_mobile_device.php file, 165

display:none, using, 36
distraction, 6–7
divs of content, displaying, 35–36

E

em versus point, considering for text, 77
“email” input type, including on contact form, 103
excitement, considering for products, 17

F

Facebook, 114
Finnish flag, resizing, 64
forms. *See also* contact form
 “email” input type, 103
 HTML5 input type, 104
 jQuery Mobile, 152–153
 labels, 108
 Phone field, 106
 Reason for Contact menu, 107
 select menu, 107–108
 “tel” input type, 106
 “text” input type, 102
 “URL” input type, 105
full_site.inc.html file, 167–168

G

Google+, 115

H

Happy Cog, 40–45
head_inc.php file, 166
heading, coding, 77
Hide and Show buttons, 86–88
Holiday Inn
 in browsers, 136–137
 search results, 134
HTML
 jQuery Mobile, 146
 mobile, 30–31
 navigation, 54–55
 versus PHP scripting language, 156
HTML elements, units and measures, 24
HTML files, modifying with PHP, 158–159
HTML5 input type, including on contact form, 104

I

images. *See also* inline images
 inline, 63–67
 photo grids, 68–72
 technique, 97
index.php page, 158
inline images. *See also* images
 cascading, 64
 enlarging, 65–67
 img tags, 63
 main.css, 63, 66
 mobile.css rule, 64
 mobilizing, 63–67
 narrowing width, 64
 sizing down, 63
input types for forms
 “email,” 103
 HTML5, 104
 “tel,” 106
 “text,” 102
 “URL,” 105
Intercontinental Hotel Group project interview, 134–138
interviews
 Arizona State University, 41–45
 Boston Globe, 92–97
 Intercontinental Hotel Group, 134–138
iOS, holding .com key in, 105
iOS app interfaces
 Calculator, 52
 iPod, 53
 Phone, 53
iPad screen, narrowest dimension of, 31
iPhone, Mobile Safari on, 29
iPod interface, 53
is_mobile_device.php script, 165–166

J

Jones, Colleen, 134
jQuery Mobile, 142, 172
 About Design in Finland screen, 147–148
 advantages, 143
 behavior, 145
 body and header bar, 146
 disadvantages, 143
 divs, 146
 fade transition, 151
 flip transition, 151
 forms, 152
 head section, 144–145

HTML, 146
landscape orientation, 153
links to JavaScript files, 145
main.css, 145
mobile.css, 145
modal dialog menu, 150–151
navigation, 147
pop transition, 151
reverse transition, 151
slide transition, 151
slidedown transition, 151
slideup transition, 151
stylesheet, 145
themes, 151
tool bar and buttons, 148–149
transition options, 151

K

Kano, Noriaki, 16
Kano index
 refining, 18
 using spreadsheet, 18
Kano Model, 16–18
 Basic tier, 17
 Excitement tier, 17
 Performance tier, 17
keyboard, responsive, 102–103, 105–106

L

labels, placing above forms, 108
Lawson, Bruce, 104
layout
 behavior of web browsers, 25
 fixed-width, 23
 mobile CSS, 32–36
 mobile HTML, 30–32
 specifying units and measures, 24
 un-fixing width, 24
Lerdorf, Rasmus, 156
LinkedIn, 117

M

main.css file
 enlarging inline images, 66
 heading, 77
 jQuery Mobile, 145
 photo grids, 69–70

 social media, 120
 using with inline images, 63
Marcotte, Ethan, 22, 31, 91–92
margins, adding, 34
media queries, 31, 171–172
mobil.css stylesheet, 32–35, 70–71
 bodycontent ID, 34
 jQuery Mobile, 145
 linking to, 30–31
 margins, 32–33
 navigation, 56–59
 padding, 32–33
 paragraphs, 79
 PHP scripting language, 169–170
 removing borders, 33
 rules in, 31
 sizing down, 64
mobile =
 busy, 4–5
 distracted, 6–7
 doing, 3
 relaxed, 8–9
 tasks, 10
mobile apps, Words With Friends, 8–9. *See also* web apps
mobile browsers, downloading content, 35–36. *See also*
 browser windows
mobile HTML, 30–31
Mobile Safari. *See also* Safari
 Design in Finland web site, 59
 on iPhone, 29, 32, 34
mobilization, iterative, 122
mobilizing
 existing web content, 35–36
 web sites as iterative process, 15
MTV O Music Awards, 41–45

N

navigation
 Calculator iOS app, 52
 car interfaces, 51
 CSS, 55–56
 HTML, 54–55
 iOS app interfaces, 52–53
 iPod, 53
 layout and behavior, 55–56
 Mobile CSS, 56–59
 #nav elements, 58
 Phone app, 53
 watches, 49

P

- paragraphs
 - `mobile.css`, 79
 - resizing, 78–80
- Phone app, 53
- Phone field, activating in mobile view, 106
- photo gallery, 68
- photo grids
 - column-count CSS3 property, 68
 - column-gap CSS3 property, 68
 - CSS3 properties, 68
 - div id, 69–70
 - `main.css` file, 69–70
 - `mobil.css`, 70–71
 - responsive, 68–72
- PHP includes, 156
 - adding, 164–165
 - CSS, 169–170
 - full site and mobile, 167–168
 - `session.inc.php`, 164
- PHP processing, 157
- PHP scripting language, 172
 - # character, 166
 - `/* comment */`, 166
 - CSS styling for buttons, 160–163
 - device detection, 165–166
 - enabling on web servers, 157
 - full screen presentation, 160
 - `full_site.inc.html` file, 167–168
 - versus HTML, 156
 - `index.php` page, 158
 - `is_mobile_device.php`, 165–166
 - `mobile.css`, 169–170
 - modifying HTML files, 158–159
 - overview, 156
 - Return to Mobile Site button, 160–163
 - single-line comments, 166
 - `test.php` file, 157
 - tooggling between presentations, 163
 - View Full Site button, 160–163
 - wrapping comments, 166
- Pivotal Tracker, using for story backlogs, 15
- point versus `em`, considering for text, 77
- products, performance requirements, 17
- project interviews
 - Arizona State University, 41–45
 - Boston Globe, 92–97
 - Intercontinental Hotel Group, 134–138
- puzzle games, layout of, 25–28

Q

- Quora, 116

R

- relaxed state, supporting users in, 9
- Return to Mobile Site button, 160–163
- rightbar div, omitting display of, 36

S

- Safari, on Mac laptop, 28. *See also* Mobile Safari
- Safari desktop, viewing interview page in, 65–66
- screen layout
 - behavior of web browsers, 25
 - fixed-width, 23
 - mobile CSS, 32–36
 - mobile HTML, 30–32
 - specifying units and measures, 24
 - un-fixing width, 24
- screen width
 - 768 pixels, 31
 - adjusting, 31
 - narrowest dimension, 31
- select menu, drop-down, 107–108
- `session.inc.php`, 164
- Sharp, Remy, 104
- Show and Hide buttons, 86–88
- Shuffle button, design update, 9
- Skype
 - design of, 6–7
 - Mute button, 6–7
- sliding tile puzzle, 25
- smartphones, relaxed state of use, 9
- social media
 - background color, 120–122
 - connecting to, 119–122
 - Facebook, 114
 - Google+, 115
 - iterative mobilization, 122
 - LinkedIn, 117
 - linking to mobile versions, 120
 - `main.css`, 120
 - mobile presentation, 120–122
 - Quora, 116
 - Twitter, 113
- solutions, requirements for, 17
- spreadsheet, using for Kano index, 18

stories

- design requirements, 14
- prioritizing, 15–16
- questions related to, 13
- usability evaluations, 13
- writing, 12–13

story backlog. *See also* user stories

- applying Kano Model to, 18
- creating, 15–18
- estimating degree of change, 18
- Kano Model, 16–18
- Pivotal Tracker, 15
- project management tools, 15

stylesheets with media queries, 171–172

subheadings, coding, 80–81

T

tablets

- relaxed state of use, 9
- use of, 8

tasks, using mobile devices for, 10

“tel” input type, including on contact form, 106

test.php file, creating, 157

text. *See also* Wikipedia

- em versus point, 77
- heading, 77–78
- line lengths, 80
- paragraphs, 78–80
- quantity, 82
- subheadings, 80

“text” input type, including on contact form, 102

Twitter, 113

U

“URL” input type, including on contact form, 105

usability evaluation, 127–129

- site message, 129
- site organization, 129

user experience

- being busy, 5
- doing things, 3
- driving around, 5
- estimating degree of change, 18

user stories. *See also* story backlog

- design requirements, 14
- prioritizing, 15–16
- questions related to, 13
- usability evaluations, 13
- writing, 12–13

users, supporting in relaxed state, 9

V

View Full Site button, 160–163

W

watches

- design, 49
- scale, 49
- size, 49

web apps, Pivotal Tracker, 15. *See also* mobile apps

web browsers, layout behavior, 25–28. *See also* browser

windows

web forms. *See* contact form; forms

web sites. *See also* Design in Finland web site

- Blueprint framework, 23
- companion, 172
- design of, 26
- jQuery Mobile, 142–143
- jQuery Mobile’s gallery of form controls, 152
- Kano Model, 16
- “Lightweight Device-Detection in PHP,” 165
- message, 129
- organization, 129
- photo grids, 69
- PHP scripting language, 156
- PHP-powered site, 171

Wikipedia. *See also* text

- Android mobile presentation, 83
- article areas, 83
- article introduction, 83–84
- behavior, 87–88
- collapse.css file, 86–87
- columns, 83
- CSS, 86–87
- full-size presentation, 82
- h3-level header, 86
- HTML, 85–86
- JavaScript, 87–88
- mobile presentation, 83–84
- Show and Hide buttons, 86
- table of contents, 84

Words With Friends mobile app, 8–9

Z

Zeldman, Jeffrey, 22, 40