O'REILLY®

The Uncertain Web

WEB DEVELOPMENT IN A CHANGING LANDSCAPE

Rob Larsen

O'REILLY®

The Uncertain Web

What's the best way to develop for a Web gone wild? That's easy. Simply scrap the rules you've relied on all these years and embrace uncertainty as a core tenet of design. In this practical book, veteran developer Rob Larsen outlines the principles of what he calls The Uncertain Web, and shows you techniques necessary to successfully make the transition.

By combining web standards, progressive enhancement, an iterative approach to design and development, and a desire to question the status quo, your team can create sites and applications that will perform well in a wide range of present and future devices. This guide points the way.

Topics include:

- Navigating thousands of browser/device/OS combinations
- Focusing on optimal, not absolute solutions
- Feature detection, Modernizr, and polyfills
- RWD, mobile first, and progressive enhancement
- UIs that work with multiple user input modes
- Image optimization, SVG, and server-side options
- The horribly complex world of web video
- The Web we want to see in the future

"A refreshingly honest look at the chaotic, wonderful world of web development, with handy, practical advice for making future-friendly, backward-compatible websites."

—Jeremy Keith
Research and Development, Clearleft

Rob Larsen has spent 13 years building websites and applications for some of the world's biggest brands. He's applied that experience to teaching a broad audience in *Beginning HTML and CSS*.

WEB DEVELOPMENT

US $29.99 **CAN $31.99**
ISBN: 978-1-491-94590-2

Twitter: @oreillymedia
facebook.com/oreilly

The Uncertain Web

Rob Larsen

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo O'REILLY®

The Uncertain Web

by Rob Larsen

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editors: Simon St. Laurent and Amy Jollymore
Production Editor: Colleen Lobner
Copyeditor: Marta Justak
Proofreader: Jasmine Kwityn

Indexer: Ellen Troutman-Zaig
Cover Designer: Ellie Volckhausen
Interior Designer: David Futato
Illustrator: Rebecca Demarest

December 2014: First Edition

Revision History for the First Edition:

2014-12-02: First release

2015-01-07: Second release

See *http://oreilly.com/catalog/errata.csp?isbn=9781491945902* for release details.

ISBN: 978-1-491-94590-2

[LSI]

Table of Contents

Preface

The best way to approach the Web today is to forgo hard-and-fast rules and design for uncertainty. Embracing uncertainty as a core tenet of web development and scrapping the rules we've relied on in the past few years is the best bet for creating future-proof web solutions.

In the early 2000s, there was basically one browser (Internet Explorer 6), one platform (Windows XP), and one screen resolution (1024 × 768) that mattered. With that setup, you could design, develop, and test the vast majority of web users with one desktop computer. The biggest question on the horizon, it seemed, was when it would be viable to design for 1280-pixel screens.

This limited field of play meant that there was an expectation that sites and applications would look the same everywhere for everyone. Best practices were honed and codified into hard-and-fast rules that drove design and development. Major choices, such as the size of the basic design grid, were no longer *choices*. Everyone started with a static, 960-pixel grid and then sliced and diced it as needed.

Today, things couldn't be more different. With the launch of the iPhone and the iPad, the rise of Android, and the growth of not just one but two real contenders to Microsoft's position as the dominant desktop web browser (Firefox and Chrome), developers and designers have an ocean of variables to navigate. Every question around a site design is now filled with options.

Initially, developers and designers tried to navigate this new reality by creating new rules. But the problem was that the goalposts kept moving. As soon as a new hard and fast rule was created, some new wrinkle would render it impotent. People designed and built iPhone sites,

assuming that Apple's dominance in the smartphone market was a permanent condition. They tested for touch capabilities and assumed that touch users would never have a mouse.

As Android's huge growth over the past few years, and the presence of Chromebooks and Windows 8 laptops with both mouse and touch capabilities have proved, those new rules have a short shelf life.

Even patterns like responsive web design (RWD), which some saw as a single solution for design and development moving forward, fell apart when applied against complicated application patterns and the questions of bandwidth and the challenge of mobile performance.

By combining web standards, progressive enhancement, an iterative approach to design and development, and a desire to question the status quo, teams can create sites and applications that should perform well in a wide range of present and future devices. By focusing on optimal solutions with intelligent fallbacks and forgoing the desire for absolute solutions, design and development can work together to create a Web that is fast, widely available, and reliable.

This book will outline both the concept and underlying principles of the uncertain Web and introduce some of the techniques necessary to make the successful transition.

A Word on the Web Today

The evolution of the Web as a development platform and the incredible growth in the number of web-enabled devices has pushed the Web into places it could never have reached before. In the past decade, we've gone from a stagnant platform with a handful of browsers and operating systems connecting to the Web to a vibrant, Open Web Platform serving a dizzying array of browsers and devices.

That's the big picture.

The thing is, most of the time, front-line developers don't get to spend time looking at the big picture. You know how it is—it's usually a challenge just getting the next release out the door. Whether you're building a site for a client, working on the latest version of your JavaScript framework, or simply trying to make sure people can read the text on your blog, there's not a lot of time available to muse about the way the Web as a whole has changed. Instead, you focus on solutions to individual problems, because those are the ones keeping you from going

home at a reasonable hour. Even folks who are tasked with keeping track of the big trends can get sidetracked by specific storms that pop up. It's hard to keep your eye on the big picture when you're watching 10 (long!) emails an hour come through on a standards topic you're following with interest.

That's where this book comes in. Judging by the conversations I see on GitHub, StackOverflow, Twitter, and IRC, it seems like people don't really think about how fundamentally the Web has changed. Whether it's searching for the perfect test to detect a "mouse" user versus a "touch" user or designing a responsive site for the "perfect" set of media query breakpoints, there are many developers still trying to hammer out absolute rules and rigid best practices. People are clearly looking to develop sites and applications within clearly defined boundaries.

Although that can be a comforting idea and was once possible, those days are long gone. It's time for a new approach.

The quote from Yehuda Katz at the beginning of the first chapter, sums up, cleverly, two of the threads that you'll see throughout the book. Flip ahead and check it out, or read it live on Twitter.

Hopefully, when you're done with this book, you'll be doing less crying.

Today's Web is a wild place. The Web has *never* been a static platform, no matter how much people might wish it were so. You just can't control who's going to request your content. You can't control the browser or device they're using, and you certainly can't guarantee things like the operating system, screen resolution, bandwidth, or available system fonts. For developers coming from pretty much any other discipline, the number of things that are out of the developer's control can be mind-boggling. That statement will only grow more true with every passing day. Standards are changing on, in some cases, a daily or weekly basis, new devices are coming online at a furious pace, and browser vendors are going at it tooth and nail to innovate their way to the top of the league tables. With an ecosystem like that, trying to collapse everything you do as a developer into something that can fit into a neat little box is a recipe for frustration. Embracing the ecosystem for the wild mess that it is and developing with an eye toward the uncertainty the Web will throw at you is the best way to reach whomever might want to get at your site or application with whatever they have in their pocket or on their desktop—now and in the future.

The tools to do this are already here; you just need an adjustment in the way you view the Web and the way you develop for it.

Who Should Read This Book

The primary audience is intermediate to advanced web developers—the folks on the front lines of dealing with these issues on a day-to-day basis and those who serve as the main channel for new frontend development techniques and trends to make their way into organizations. This book is geared toward developers who work primarily in HTML, CSS, and JavaScript and who have a solid understanding of cross-browser (if not cross-form-factor or cross-device) development techniques.

The secondary audience consists of user experience designers, web-focused visual designers, and web-focused engineers from other programming disciplines. To properly build for the modern Web, there needs to be cohesion in site design and architecture from start to finish. The material here should familiarize other disciplines with the best way to approach designing and developing for the present and future of the Web. As a natural bridge between design and the server, the core web developer is always going to be the glue that binds this process together, but having everyone on board will help improve the finished product.

Navigating This Book

This book is organized into three parts.

Chapter 1 and Chapter 2 will establish the current environment we're in and show us how embracing uncertainty and building for the Web as it exists is the way to go. If you've been handed this book by a coworker and you're not particularly technical (at least in terms of frontend development), then these two chapters (and the conclusion) are the ones to read.

The next several chapters present some of the technical challenges we're facing on the Web and illustrate some of the ways that embracing uncertainty can help solve them. Chapter 3 provides a quick introduction to Modernizr and feature detection (techniques to enable or disable functionality based on browser support). From there, we move on to responsive web design (Chapter 4), user input in the current multidevice landscape (Chapter 5), images on the Web (Chapter 6),

and modern web video (Chapter 7). If you're interested in a way to think about this stuff technically, then you'll want to read these chapters. Each one is split between examining the true scope of a problem on the modern Web and looking at solutions and how embracing uncertainty can help you reach the widest possible audience.

I didn't plan it this way when I started the book, but the last three technical chapters (Chapters 5–7) represent the full spectrum of success and failure in the standards process. Video represents a good standard gone bad, user input represents a problem still in search of a solution after several years of development, and images represent a success created by the entire technical web community (standards authors, web developers, and browser vendors).

Chapter 8 takes a final look at the uncertain Web and then talks about the Web as I want to see it evolve over the next 20 years.

Online Resources

The following sites are where modern web design and development are being figured out. These sites have all directly influenced the development of this book:

- HTML5 Rocks—A resource for Open Web HTML5 developers
- LukeW Ideation + Design | Digital Product Strategy & Design
- QuirksMode—for all your browser quirks
- Web Hypertext Application Technology Working Group
- The Modernizr issue tracker on GitHub
- CSS Tricks
- A List Apart: For People Who Make Websites

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

`Constant width italic`
: Shows text that should be replaced with user-supplied values or by values determined by context.

This icon signifies a tip, suggestion, or general note.

This icon indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *https://github.com/roblarsen/the-uncertain-web*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of

examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*The Uncertain Web* by Rob Larsen (O'Reilly). Copyright 2015 Rob Larsen, 978-1-491-94590-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://bit.ly/uncertain_web*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

I'd like to thank all the people who put up with me pestering them for feedback while I was developing this idea into a proper book. In particular, Paul Irish, Bob Holt, Marc Neuwirth, and Adam McIntyre all provided great feedback on the concept and the title. Thanks to you guys, I felt like I was actually onto something. Bob and Adam deserve double thanks for their excellent technical (and beyond) feedback throughout the writing process.

Everyone from O'Reilly has been great throughout this project, and I want to wrap the whole company up in a bear hug for that. I especially want to thank Simon St. Laurent for taking an idea sketched out in a few paragraphs and pitched at the airport in Denver and then turning it into the book you're reading now. I also have to thank Amy Jollymore

for shepherding me through this project with welcome positivity and great insight.

I'd also like to thank Lynn Haller from Studio B for getting this concept into O'Reilly's hands in the first place and for taking care of the business end of this whole book-writing thing.

Finally, I'd like to thank my wife for her love and support. I'm always busy doing something silly like writing a book, and she's always there for me. I couldn't ask for anything more.

CHAPTER 1
Embracing Uncertainty

> The web platform is Write Once, Cry Everywhere.
>
> — Yehuda Katz

I love the Web. I've been making sites for a living since 1999, and I still love the work as much as I did in those crazy days. I'm not sure how many other folks I know who can say the same thing about their profession. Admittedly, the Web has been very good to me. I've been able to travel the world, have written a bunch of articles and a couple of books, and have paid my bills with nothing but a keyboard for the past decade and a half. The thing is, while all that is great and I thank my lucky stars that I've had this career, what I really love about the Web is that it made good on its early promise. It might have sounded a little hokey or looked like just hype to fill a five-minute slot on the evening news, but the Web really has managed to connect people in incredible ways—ways we couldn't even have imagined 25 years ago. Individuals who would never have had a voice can now broadcast to the world with blogs, YouTube, Twitter, and Facebook. Politicians, filmmakers, video game developers, and anyone else with an idea can tap into the power of individuals to finance their dreams, five dollars at a time. Lessons from the world's great universities like Stanford and MIT, as well as lessons made directly for the Web from organizations like Khan Academy, are available for free to anyone in the world who can connect to the Web. With sites like GitHub, taking part in open source software is as easy as firing up a web browser and finding a place to help out with even the most massive open source projects like jQuery, Node.js, or Ruby on Rails.

It's only getting better. As more and more people come online, they're exposed to these same opportunities and start to feed back into the system with a unique voice: hard work on some open source bug, adding to the coverage of breaking news (say, sharing a photo of a plane landing on the Hudson River), or something as simple as buying a business cat tie on Etsy and turning the wheels of commerce.

It's really pretty cool.

I could go on about this for a while and, if I didn't have other plans, I'd be tempted to do just that. I *do* have plans though, so I'm going to resist the impulse.

This chapter will introduce the core concept of uncertainty in the context of web development. From there, we'll look at where we came from with Microsoft's monoculture in the early 2000s, and then we'll look in depth at where we are today. Throughout, we'll look at what the factors that have gotten us here can teach us about the future of the Open Web Platform and related ecosystem.

You won't be an absolute expert on everything the Web has to offer just by reading this chapter, but you should have a much better sense of where we've been, where we are, what's on the horizon, what some current issues are, and what kinds of things might surprise us in the future.

Embrace Uncertainty

Along with the landscape, the general philosophy of making websites and applications has slowly shifted over the past decade or so. It's moved gradually from the rigorously defined boundaries of the Microsoft monoculture to the fluid environment we have today. Design approaches like responsive web design, technology patterns like progressive enhancement, and libraries like Modernizr are all much better suited to today's Web than anything that came before. Fixed-width sites with "best viewed with" banners that broke without third-party plug-ins like Adobe Flash or failed to function if the user visited with a new, novel web browser (no matter how powerful) don't have to exist anymore. We're better than that.

That's a good start.

The thing is, although we've mostly shifted away from static 960px grids and all of the other baggage that came with the limited universe,

the shift has been isolated to islands of innovation and has generally only happened in reaction to outside stimuli. Every change in the browser and device landscape has sent people scurrying, trying to solve problems caused by new features, browsers, or form factors. Although there have been some truly flexible solutions crafted for these issues, there are just as likely to be a newly revised set of inflexible guidelines put up, only to be revisited the next time the landscape shifts. It's time to get ahead of the curve and do our best to cure the disease and not just treat the symptoms.

It's time to embrace uncertainty.

Embracing uncertainty means that we need to make the final leap away from the search for absolutes in order to appreciate the Web for what it actually is. As we'll examine in this chapter, it's a place where a wide range of devices running a wide range of web browsers in the hands of many different kinds of people are all trying to find their way to something that matters to them. Whether it's a farmer in Africa trying to figure out the score of the latest Manchester United match, a banker in Hong Kong trying to get a price for Bordeaux futures, or a small business owner in the United States setting up an Etsy shop, the Web is making important connections for people, and we need to help them on their way.

So how to do it?

We'll start by looking at specific recommendations beginning with the next chapter, but even before you start to look at the particulars, you can start to change the way you look at the process.

The initial step is to understand, from the second that you start a site design, that you (*probably*) can't control what devices, browsers, and form factors will be ingesting your content. Even better, if you can let go of the *desire* to control what devices, browsers, and form factors are accessing your site, you'll be even happier with your results. While some organizations and certain applications *can* dictate specific browser and OS versions, you're probably not going to be able to do so on your end. Unless you're in control of your users' machines or are offering something no one can get anywhere else, you should be trying to satisfy as many browsers and devices as you possibly can.

"But My Client Doesn't Care"

A common issue when people start to embrace anything new in the web development sphere, whether it's a formal usability program, an accessibility initiative, or a web performance optimization project, is getting buy-in from clients or internal stakeholders who might not immediately understand the benefit of something new and unfamiliar. I imagine much the same reaction to the concepts present in this book. I've certainly seen my share of pushback when sharing some of the ideas I'll be discussing, so I expect other people will see the same thing. All you can do is do what I've done: make your case with enthusiasm and data to back it up. You can't force people to change their ways, but if you present good data with conviction, you've got a better chance than if you sit idly by and do nothing.

The next, most important step is not only to accept that you can't control the browser and device environment, but to embrace the ecosystem for what it is—a sign that the Web is a healthy platform. Tens of thousands of browser/device/OS combinations is a *feature* of the Web, not a problem.

I talk to a lot of people, and there are plenty of complaints about the Web as a platform. I'm not talking about specific complaints about specific features. I'm talking about complaints about the Web itself.

Many Java/C/C++ developers just shake their heads at the idea that code written for the Web can be executed in so many different environments and can have just as many different results. To them, the Web is just nuts. On the other end of the spectrum, many web developers have their favorite browsers, great hardware, new smartphones, and everything else gets the short end of the stick. These are the folks who go over the top in GitHub issues with their hatred of Internet Explorer, test 99% of the time in Chrome, and are actively wishing for WebKit to be the only rendering engine on the Web because it would make things so much easier for them.

Don't be either extreme.

Instead of worrying about the fracture in the Web and wishing that it was something else, accept the Web for the blessing that it is. And it is a blessing. Because the core technology can run, unaltered, on billions of devices in the hands of billions of people, you have immediate access to all of those billions of people and all of those billions of devices.

How great is that?

From Microsoft's Monoculture to Today's Healthy Chaos

In the early 2000s, there was basically one browser, one platform, and one screen resolution that mattered. You could test the experience of the vast majority of your users, with excellent fidelity, simply by running Windows XP with Internet Explorer 6 and switching between a couple of different screen resolutions (i.e., 800 × 600 and 1024 × 768 pixels). Add in Internet Explorer 5 and 5.5, and you could hit, by some estimates, more than 95% of the Web. In the end, Internet Explorer held market share of near or above 90% for most of the first half of the 2000s.

To Be Fair, Internet Explorer Was the Good Browser in 1999

Whatever you might think of their business practices at the time (and the courts certainly didn't take kindly to them), if you had to choose to *develop* for any browser in the late dot-com era, it was going to be Internet Explorer. Far from being the butt of jokes, Internet Explorer versions 4 through 6 were, at the time, each the best available browser by a wide margin. I've said it many times, and I'll say it again here, the worst major browser ever was Netscape 4. Internet Explorer 6 may have overstayed its welcome by about seven years, but Netscape 4 was simply born bad.

What's more, beyond simply being the most powerful browser, the Internet Explorer team consistently pushed out powerful features and APIs that still resonate on the Web today. For one example, the XMLHttpRequest object, which serves as the foundation of modern frontend development, was an Internet Explorer innovation. It really doesn't get any more important than that, in terms of single innovations that have changed the way that we architect web solutions.

For more perspective on what Internet Explorer brought to the Web in those early days, check out my blog post, "Some Internet Explorer Innovations You Probably Forgot About While Waiting for IE6 To Die" and Nicholas Zakas' blog post, "The Innovations of Internet Explorer."

Slowly, from the height of Internet Explorer's dominance (reached in the middle of 2004), things began to turn. It really started with Firefox, the heir to the Netscape mantle, chipping away at Internet Explorer's dominance by presenting an independent, standards-compliant alternative. With Opera revamped for modern development in 2003 (it had previously been great for CSS and *weird* for JavaScript), the 2003 release of Apple Safari, and 2008 release of Google's Chrome browser, Internet Explorer had real competition on multiple fronts, each taking a chunk out of the giant until it was eventually toppled as not only the dominant browser version, but the dominant browser family in May 2012.

What's more, while all that desktop competition was heating up, an entirely new front in the browser wars opened up with the unprecedented growth of the mobile Web. With the launch of the iPhone and iPad and the dominant growth of phones powered by Google's Android operating system that followed, both the absolute number of users and the number of devices used to connect to the Web per user grew.

Additionally, browser vendors have almost universally (with Apple being the only holdout) instituted a policy automatically pushing updates. Gone are the days of new browser versions shipping every couple of years alongside a new OS update. This new commitment from the browser vendors has allowed us to add new web platform features at a breakneck pace. It has also led to a spread of browser versions, as different organizations and individuals move to the latest version at their own speed.

So, instead of having a couple of machines dedicated to testing and getting 95% coverage, anyone who really pays attention to this stuff can have a testing lab with 50 or more devices and still struggle to cover the same high proportion of the Web that was possible during Microsoft's heyday.

If Hollywood were going to do an edgy reboot of the "Rip Van Winkle" story, they might as well use a web developer, because a developer taking a nap under his desk (as I often threaten to do) in 2004 and waking up today would be bewildered by the changes in the landscape. Imagine a cockeyed Owen Wilson asking, "Google has a browser?" There'd be a lot of that kind of thing.

I mean, I've been paying close attention the whole time, and the changes are just nuts to me.