Responsive Design
High Performance

Creating a speedy, user-friendly, and uniform experience across different devices has been a challenge with responsive design. 72 percent of responsive websites deliver the same volume of bytes irrespective of the device or screen size, which makes it difficult for sites to load on slow websites.

We will start with an overview of all of the workable aspects of responsive design, and then we will quickly focus on practical things that we can start doing without much effort to get our sites performing better than ever before on different types of devices. We will then get into one of the most important aspects of websites: managing images. Since we need our images to display nicely and also load quickly across all devices such as computer, mobile, tablet, and so on, we will look at ways to do so in this book.

Who this book is written for

This book is ideal for developers who have experience in developing websites or possess minor knowledge of how responsive websites work. No experience of high-level website development or performance tweaking is required.

What you will learn from this book

- Develop good habits when placing DOM elements, resources, and become familiar with DNS-Prefetching
- Understand how images affect your site’s performance and learn the limitations
- Get to grips with style sheets, media queries, and viewports
- Take control of HTTP requests and learn how to keep them to a minimum
- Understand the origin of responsive design
- Improve your development time by learning about new tools
- Discover some in-browser developer tools and learn how they help gauge performance

Responsive Design
High Performance

Leverage the power of responsive design to fine-tune your website’s performance and increase its speed

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'The Good, the Bad, and the Ugly of Responsive Web Design'
- A synopsis of the book’s content
- More information on Responsive Design High Performance

About the Author

**Dewald Els** is short, dark, hairy, and curious. He has extensive experience in PHP and JavaScript. He has worked in the corporate sector after some experience in video game development in C#.

After working in video game development, Dewald joined one of South Africa's top three ISP service providers. He was in the team that developed a solid backend for the ISP from the ground up, to better serve clients. After moving to Pretoria, he currently works for Vane Systems, maintaining their event sales website, [http://www.ticbox.com](http://www.ticbox.com). He takes the lead in developing new features for the site.
Responsive Design
High Performance

Is it responsive? With a growing number of hardware devices that access the web, building a website that is responsive has almost become expected. This can be both good and bad, as responsiveness is a daunting task when not understood properly. So, let's talk about it.

What is responsive design, anyway?

Simply put, a responsive website adapts its layout according to the device being used to access it. The website will dynamically change its layout depending on the device's screen size and orientation.

Some examples of this include menus that collapse, images that resize, and column structures rearranging (for example, a two-column layout converging into one column).

The goal of this book is to make designing and building a website less of a daunting task for the ever-growing number of devices that we use to obtain information from the Internet. We access the Internet from so many different devices, ranging from smart watches to mobile phones and desktops with large high-resolution displays, that responsiveness has become an almost essential feature in web design and development nowadays.

There are prerequisites for a site to qualify as a responsive website. Fluid grids, media queries, and flexible images are a few of them. Additionally, a distinction does need to be made between an adaptive layout and a responsive layout.

Adaptive Layout

An adaptive layout uses fixed-width grids or columns that are triggered at specific and static points.

Responsive Layout

A responsive website makes use of fluid grids; in other words, the grids resize as the viewport size changes.
What This Book Covers

*Chapter 1, The Good, the Bad, and the Ugly of Responsive Web Design,* covers what responsive design is and, more importantly, *why* it's so important on the Web today.

Responsive web design helps us create a more uniform appearance across an array of devices. It leaves users feeling more familiar with a brand, regardless of the device they're using to interact with it. Code is kept together in one place, and this negates the need to maintain multiple pages or documents for one website.

*Chapter 2, Tweaking Your Website for Performance,* describes the importance of resource placement on the DOM. It's true that placing scripts at the bottom of the page improves performance, but it's also true that some scripts need to load before the DOM renders.

Preloading content can be greatly helpful in improving the initial load of your site. DNS prefetching is a very helpful piece of code that can resolve the DNS name in the background for the site that your page might point to. Search engines can greatly benefit from this.

*Chapter 3, Managing Images,* proves that managing images is no small feat in a website. Almost all modern websites rely heavily on images to present content to the users. This chapter explains which image formats to use and when, and also how to optimize your images for websites. We discuss the difference between progressive and optimized images. Conditional loading can greatly help us load our site faster, and we discuss how to use conditional loading to improve our site's performance. We touch on server-side optimization using caches, ETags, and media queries for retina displays.

*Chapter 4, Learning Content Management,* takes your development in the right direction. We take a look at style sheets, media queries, and how to work with viewports. This chapter also covers the use of CSS preprocessors such as SASSY CSS, SASS, and LESS. Though these are not covered in detail, this serves as an introduction to help you get started. CSS preprocessors are an excellent way to code your style sheets; once you have the hang of it, you can eliminate lines of unnecessary code.

The Web is slowly but surely moving towards an app-like experience, and frameworks such as AngularJS are right at the forefront of this movement.

This chapter also covers conditional content management with the `<link>` tag to load style sheets based on device size requirements.

Optimization doesn't just occur on the client side, but sometimes on the server side as well, to help deliver correct content before it reaches the client. This is an effective way to manage content.
Parallel downloads are briefly covered. We discuss the fact that the time that the client spends waiting for requests to finish is referred to as blocking, and that our goal should be to reduce blocking as much as possible to achieve shorter load times.

We also cover another excellent way to deliver your sites quickly: by making use of content delivery networks. Then we discuss fonts and how they can affect a website's performance.

Chapter 5, *The Fastest HTTP Request is No HTTP Request*, proves that one of the best ways to improve the load time of your website is by reducing requests. We look at some effective and easy-to-implement techniques to help you achieve speedy load times. We discuss sprite sheets and how they can take a bunch of requests and turn it into one request. We also provide information about combining files.

Server-side optimization is also a great place to make some improvements. We talk more about server-side optimization and also take a look at AppCache, which is another excellent method to improve your site's performance.

Chapter 6, *Testing, Testing, and Testing*, asserts that testing is a crucial stage of the development life cycle. It's where you can not only identify problems with your site but also root out performance issues.

Chapter 7, *Speeding Up Development with Design Concepts, Patterns, and Programs*, focuses on a few ways to improve your site. We kick off by looking at design concepts: graceful degradation, and progressive enhancement. We discuss the differences between these two concepts and how a better user experience can be achieved on a website by making use of progressive enhancement rather than graceful degradation.

Making use of object-oriented CSS (OOCSS) can be a great benefit to your website's maintainability and can improve its loading speed. By following OOCSS, you can reduce the size of your CSS files, thereby improving the download time of the resources required to load your site. We also take a look at how we can improve OOCSS even further by combing it with a CSS preprocessor such as SASS.

There is a brief mention of available patterns to get you started with your project. Programs such as GruntJS and RequireJS can make you more productive and improve your website's performance. We describe them as well.
Chapter 8, Using Tools for Performance, is basically an introduction. It is intended to point out available applications that you can use to make your development cycle more efficient.

Once we're almost done, we take a look at some resources to take what we've talked about to the next level. We mention some great developers who have made significant contributions to the web field and give you some excellent starting points to further your knowledge of responsive web design.

Appendix, Taking the Next Steps, contains an overview of the entire book. It summarizes what we talked about and helps you understand the next steps to becoming a great, responsive web developer.
Responsive web design, often referred to as RWD, has brought many great things for web designers since its inception in 2004, although the term was only coined in 2007 by Ethan Marcotte. The technique of adapting the layout of a site was written by Cameron Adams in 2004.

Here is a breakdown of what we'll discuss in this chapter:

- An overview of the good, the bad, and the ugly of responsive design
- We will look at some examples in each case
- Thereafter, we will take a look at the effects of each example, and how it affects end users and the business
The good – appearance and management

We will be going through the good aspects of responsive web design in the following sections.

Appearance

In appearance, these are the aspects that really stand out:

- **Conformity**: One of the great advantages of responsive web design is the conformity that it brings to our ever-growing, multidevice, browsing experience online. Modern web pages can now easily carry the same design characteristics from desktop to tablet and even to mobile browsers without compromise, thereby greatly enhancing a brand's web presence. Another perk is the ease that this approach brings to code maintenance. In the following screenshot, we can see a good example of adapting a site for multiple devices:
• **User interaction**: Simply changing the site to fit inside different devices is, of course, only the tip of the iceberg. With each change of the layout, the website's usability must remain intact or, in some cases, change to suit the device that it's being viewed on. Here is a good example of user experience staying consistent throughout, from desktop to mobile:

![Desktop website layout](image)

- **User interaction (desktop layout)**: The preceding screenshot shows the desktop version of a website that has a full menu, with functions such as **Sign In** and **Follow us** that are easy to access. The content is well spaced and feels clean. The column space on the right is used for calling action links that show some of their products. The column space on the left is reserved for more involving content, with images and headings.
Let's compare this to the mobile layout, which is shown in the following screenshot:

- **User interaction (mobile layout):** In the mobile view, we can clearly see how the designers have made space for the content to take center stage on the site. The navigation collapses to show easily recognizable icons, and the main content takes up the rest of the page space, which is perfect for mobile phones.
• **Appearance** (focusing on content): When it comes to mobiles, content takes center stage. Studies indicate that some users leave a site after merely 3 seconds if the content has not loaded. Responsive web design puts the focus on content. When a mobile site loads, the content needs to be easy to find and should not force the user to scroll endlessly to find what they’re looking for.

**Management**

Here are the benefits of responsive web design from the management perspective:

- **One code source**: Responsive websites have the advantage over the old mdot way of developing by virtue of keeping all of your code in one place. Another tremendous advantage of having one code source is that it avoids multiple redirects to an mdot web application. Redirects are very expensive in terms of load time and could add significant time to it.

- **Easier to maintain and update**: Besides one source code, the next obvious advantage is code that is easier to maintain. With all of the code centralized, it becomes a much less demanding task to keep all your sites up-to-date. One change on your desktop site will automatically reflect on both the tablet and mobile versions, without any extra development time.

**The bad – slow load times and unresponsive interactions**

Like most things, with the good comes the bad, and responsive web design is no different. Without proper optimization and careful planning, your responsive website could be slow and painful for the end user to navigate. Conscientious efforts to optimize the end user’s experience are an integral part of good responsive design and development.

**Slow load times**

One of the biggest culprits when it comes to slow load times is images. All too often, the same-sized image used on the desktop site will be loaded for the mobile version as well. This is considered bad practice; when it comes to mobile browsing, every kilobyte counts. So, why let a user download a 300 kB file when they only need to download a 100 kB file? Creating appropriately sized images for various devices is a must.
Let's take a look at an example. The next two screenshots show a comparison of image downloads between desktop and mobile versions:

In the preceding screenshot, you can clearly see that the image downloaded is **1140** pixels wide by **641** pixels high. This is a fairly standard header image size for a desktop site to download. Now let's see what happens when the site is viewed at mobile size, as shown in the following screenshot:

In the mobile view, the website still looks great. The image is **385** pixels wide by **216** pixels high, but take note of the natural size of the image displayed. The natural size of the image is still **1140** pixels by **641** pixels. This means that the same image was downloaded to be displayed on the mobile website as the desktop layout. This might not seem like a big deal, but the experience of waiting to download an image of that size on a mobile device could very well lose you viewers on your website.

We will cover some great ways to avoid this problem a bit later. There are some excellent techniques available to manage your image downloading based on your current screen size.
Browser requests

Another cause of slow load times is the number of requests that your browser is making. Limiting the number of requests made to your server to download content, style sheets, or scripts will greatly improve your page's load times.

Using techniques such as minification to reduce the size of the response also goes a long way towards making your website load incredibly fast.

As an example, I've included the sizes of the two style sheets from the Bootstrap framework—bootstrap.css and bootstrap.min.css. The latter is minified, and the former is not.

Take a look at the size difference between the two files, as shown here:

<table>
<thead>
<tr>
<th>File</th>
<th>Type</th>
<th>Size</th>
<th>Minified</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>bootstrap.css</td>
<td>Stylesheet</td>
<td>304</td>
<td>Yes</td>
<td>129.44 KB</td>
</tr>
<tr>
<td>bootstrap.min.css</td>
<td>Stylesheet</td>
<td>304</td>
<td>Yes</td>
<td>106.95 KB</td>
</tr>
</tbody>
</table>

The file that has not been minified has added almost 24 kB to the request. That might not sound like a whole lot, but there are multiple requests for JavaScript files, cascading style sheets, and other scripts going on at the same time, and it'll all add up.

We will take a look at this a bit later, and discuss how to implement some simple code to reduce the number of requests made; we'll also explore which tools we can use to get our code minified.

Unresponsive interactions

Clicking on a button and not seeing an immediate response from a user interface can be terribly frustrating.

A website that is not optimized and downloads unnecessary JavaScript files and bloated HTML documents (among other things) is prone to performance issues when it comes to interaction with the server.
The ugly – the effects of slow performance

If you've ever opened a website and had to wait an inordinate amount of time for it to load, I don't have to tell you that it can become an annoyance. Not only do people disassociate from the brand, but they become frustrated and would rather try to find the content they want from a faster, more reliable source.

Effects on the end user

As mentioned before, we have mere seconds to engage the viewer.

Take this excerpt from a study done by KISSmetrics from the article, How Loading Time Affects Your Bottom Line, by Sean Work, as an indicator. For further information, please visit https://blog.kissmetrics.com/loading-time/

![PATIENCE OF MOBILE WEB USERS](image)

How long are users willing to wait for a site to load before they abandon the page? The following graph seeks to answer this question.

Observation: Most participants in the survey would wait 6-10 seconds before they abandon pages.
The preceding screenshots gives us some interesting facts:

- 73 percent of mobile Internet users say that they've encountered a website that was too slow upon loading
- 51 percent of mobile Internet users say that they've encountered a website that crashed, froze, or returned an error
- 38 percent of mobile Internet users say that they've encountered a website that wasn't available
- 47 percent of consumers expect a web page to load in 2 seconds or fewer
- 40 percent of people abandon a website that takes more than 3 seconds to load
- A 1-second delay in page response can result in a 7-percent reduction in conversions
- If an e-commerce site is making $100,000 per day, a 1-second page delay could potentially cost the company $2.5 million in lost sales every year

**Effects on business**

Improving a website could have a greater impact on your business than you may realize. This is a quote from an article at [http://www.getelastic.com/](http://www.getelastic.com/):

"Walmart used a mix of pre-design, hands-on usability testing including paper-prototypes with post-design user tests (using moderated sessions throughout Canada) and on-site A/B testing, including an initial test of running both the responsive and non-responsive sites concurrently for about a week.

Results were very positive for the responsive design. Conversion’s up 20%, mobile orders up 98%.


---

[9]
Let's take a look at this simple but informative image detailing the effects that load times have on consumers. It's clear that slow load times have a tremendously negative effect on consumers who shop online:

**HOW WEBSITE PERFORMANCE AFFECTS SHOPPING BEHAVIOR**

- **47%** of consumers expect a web page to load in 2 seconds or less.
- **79%** of shoppers who are dissatisfied with website performance are less likely to buy from the same site again.
- **16%** - A 1 second delay (or 3 seconds of waiting) decreases customer satisfaction by about 16%.
- **40%** abandon a website that takes more than 3 seconds to load.
- **52%** of online shoppers state that quick page loading is important to their site loyalty.
- **44%** of online shoppers will tell their friends about a bad experience online.
Summary
In this chapter, we covered briefly what responsive design is, and more importantly, why it's so important in today's webscape.

Responsive web design helps us create a more uniform appearance across an array of devices, and leaves users feeling more familiar with a brand, regardless of the device they're using to interact with it.

Code is kept together in one place and negates the need to maintain multiple pages or documents for one website.

Code that has not been optimized and reckless content download can cause websites that look great to feel terrible. This could potentially cause users to leave the site before it's even done loading. In some cases, it may even cost you money. With all of that out of the way, let's get practical and start improving our responsive website's performance. We'll look at the placement of our resources, how to avoid common mistakes with `<image>` tags, and some other great techniques that we can apply instantly to see an improvement in website load times.
Where to buy this book
You can buy Responsive Design High Performance from the Packt Publishing website. Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers. Click here for ordering and shipping details.