Chapter 1

Why Web Performance MATTERS FOR DESIGNERS

Speed is one of the most overlooked aspects when it comes to websites and web apps. Today, users expect a website to load in 2 seconds or less.

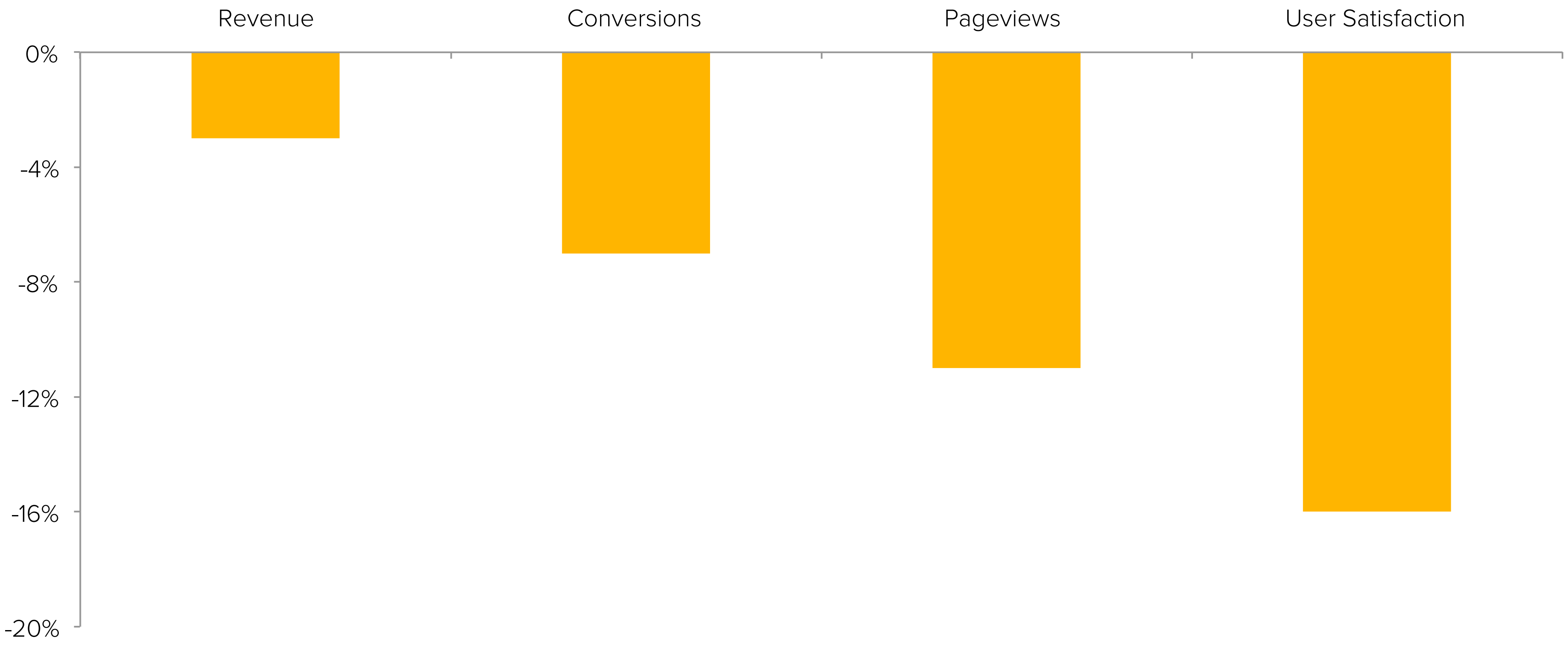
In fact, a 1-second delay in page load time equals a 3% drop in revenue per visitor, 7% fewer conversions, and a 16% decrease in customer satisfaction. When it comes to web performance, every millisecond counts.

> Speed is the ***most important*** feature. If your application is slow, people won’t use it.
>
> ***Fred Wilson***, *Managing Partner @ Union Square Ventures*

So, why should designers care? Because 80%-90% of performance occurs on the front-end of websites, which is where designers operate. Only 10%-20% happens via back-end performance, which falls under the responsibility of developers and operations. In other words, all of the content a designer is responsible for creating and building - HTML, CSS, JavaScript, images, etc. - accounts for almost all of a page's load time.

A faster website or web app will produce better business results and a better user experience. It will significantly increase visitors, conversions, average order size, and revenue both in the short and long-term. There are endless case studies proving the importance for web performance.

AVG IMPACT OF 1-SECOND DELAY IN PAGE LOAD TIME



CASE STUDY #1

amazon.com®

100ms DECREASE IN PAGE LOAD TIME

=

1% DROP IN TOTAL REVENUE

Amazon found that a 100ms delay in page load time caused a **1% drop in total revenue** - that's millions of dollars for them!

CASE STUDY #2

shopzilla

12%

INCREASE IN REVENUE

+

25%

INCREASE IN PAGEVIEWS

Shopzilla sped up its average page load time from 6 seconds to 1.2 seconds and experienced a **12% increase in revenue** and a **25% increase in pageviews**.

CASE STUDY #3

mozilla Firefox®

15.4% = **60 million**

INCREASE IN CONVERSIONS

MORE DOWNLOADS PER YEAR

Mozilla shaved 2.2 seconds off their landing pages, increasing their download conversions by 15.4%, which will result in **60 million more Firefox downloads per year**.

Chapter 2

Why Designers Struggle to **BUILD FAST WEBSITES**

There are several key reasons why building a fast website or web app has become so challenging for designers.

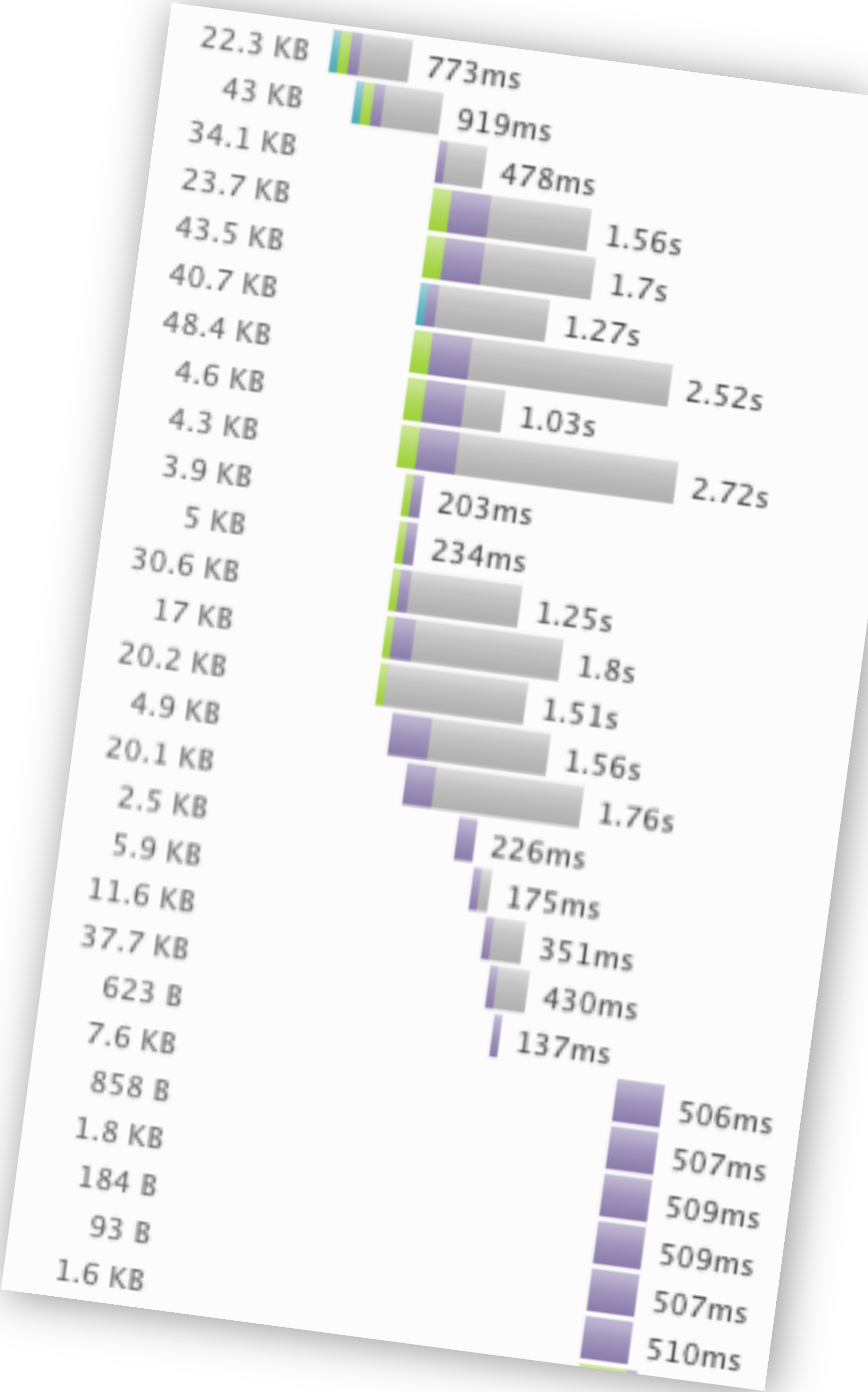
First, creating and building content by nature decreases performance. The more content a webpage has, the worse it will perform. As Steve Souders, legendary web performance optimization guru, once said, "The fastest HTTP request is the one not made."

> The *fastest* HTTP request is the one not made.
>
> ***Steve Souders***, *Head Performance Engineer @ Google*

So, the best way to optimize web performance, in theory, is to have no content at all. But, users now expect websites to contain more content, media, and interactivity than ever before. Striking a balance of fast performance and content-rich sites is one of the biggest challenges to running a great website today.

Second, many designers are unaware of the performance consequences that their assets create.

They lack the visibility and insight into their web performance issues - for example, knowing that a specific JavaScript file is slowing down their page load time by 3 seconds. They don't have or use the tools necessary to assess their web performance. Additionally, many designers do not know what steps they must take to fix these performance problems - such as moving that specific piece of JavaScript to the bottom of their HTML code to prevent it from blocking the rest of the page downloading.



Lastly, without the help of a third-party service, the methods and techniques needed to optimize a website are very manual and time-consuming efforts.

With tight deadlines and limited cycles, many designers do not take the time necessary or make it a priority to build a high performance website. Maintaining fast and consistent performance is a highly complex and demanding job. As a reference, Google, who is renowned for their fast performance and user experience, has a team of over 100 engineers whose sole job is focused on web performance optimization!

Chapter 3

What Causes Poor **WEB PERFORMANCE?**

What are the main culprits of poor performance? What components on the front-end cause the biggest problems?

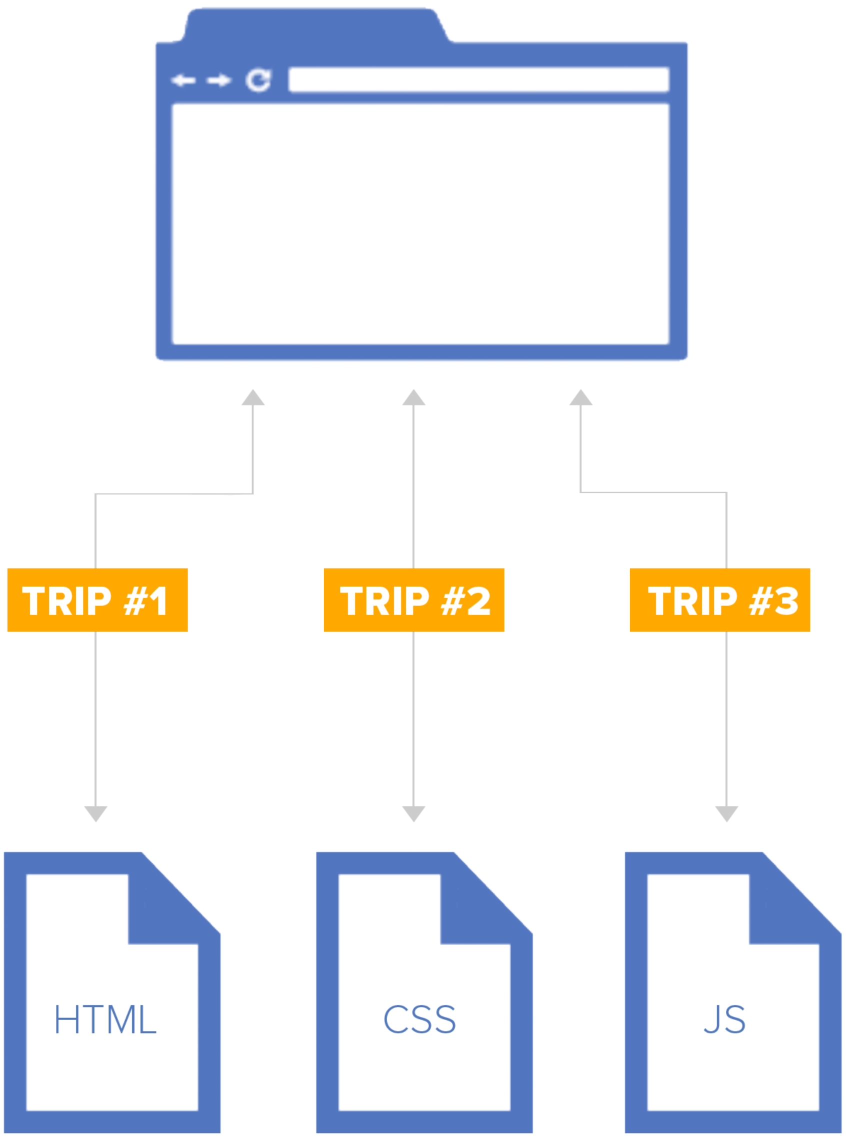
The key area we are focusing on improving performance in this eBook is a website's content complexity. Designers should mainly be concerned with optimizing assets such as HTML, CSS, JavaScript, and image files.

> If your site is 15MB, it’s not HTML5 – it’s stupid.
>
> ***Christian Heilmann***, *Principal Developer Evangelist @ Mozilla*

As we previously alluded to, a high number of files or assets is the biggest indicator of poor performance. Reducing the number of round trips and requests the browser needs to take to download a webpage will make the biggest impact in improving web performance.

Picture yourself going to get a bunch of mail from your mailbox.

In this example, you are like the browser, having to make trips to retrieve all of the assets needed to download your webpage. Would it make sense to make a separate trip to get every single piece of mail in your mailbox? No. But that's how a browser is programmed to behave when it downloads assets of a webpage – it makes a separate trip for every individual file. Granted, the browser does this very quickly - often, in milliseconds. But the more individual files there are, the longer it will take the browser to process all of those requests.



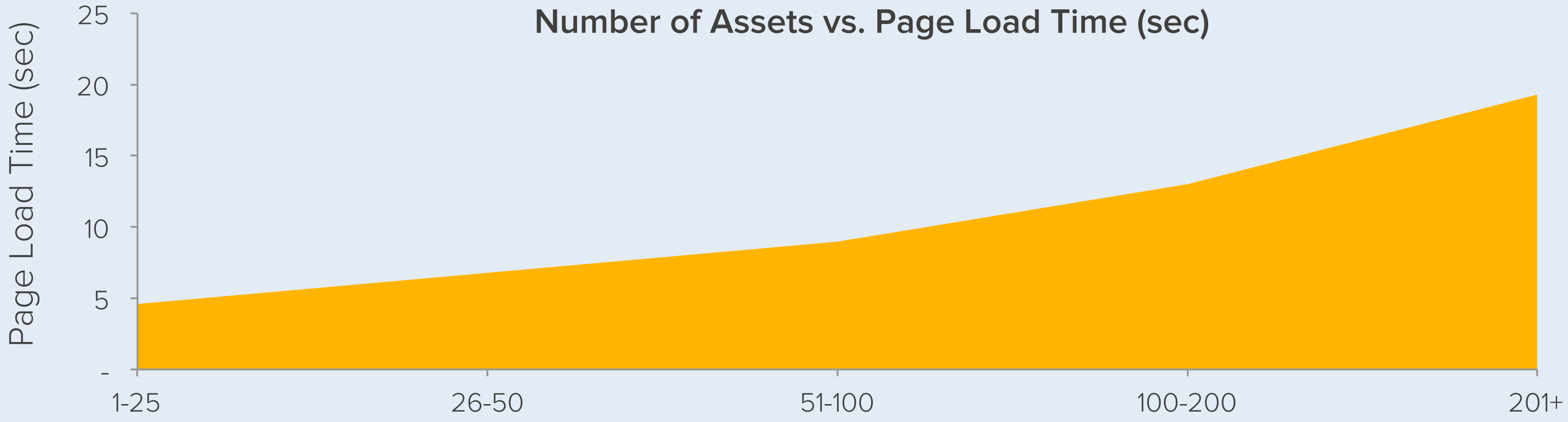
# OF ASSETS BREAKDOWN

59

AVG # OF ASSETS

AVG # OF IMAGES	**33**
AVG # OF JAVASCRIPT FILES	**11**
AVG # OF CSS FILES	**5**
AVG # OF OTHER FILES (VIDEO, AUDIO, ETC.)	**10**

When optimizing for performance, designers should reduce the number of files above all else as the number one priority. The key method to reduce the total number of assets on a webpage is combining files.



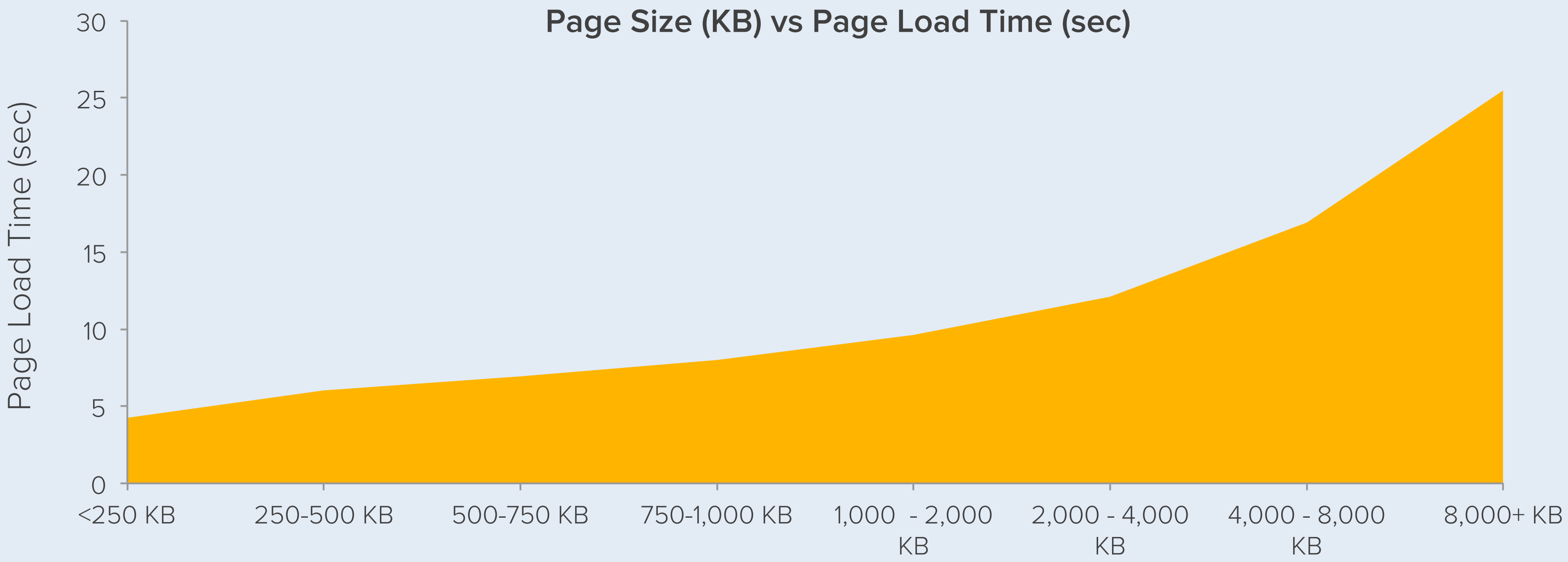
Additionally, large page size is another key indicator of poor performance. Bigger files take more time for browsers to download. Let's return to the mailbox analogy we just used. It's easy to carry a few envelopes of mail. But what if there were several heavy boxes and packages? All of a sudden, you won't be able to carry your mail as quickly or easily. The same holds true for browsers. If you have massive files on your website, browsers will have a tougher time downloading and delivering them to end users. So, designers need to compress and reduce file size as much as possible to achieve the fastest page load time possible.

ASSET SIZE BREAKDOWN

1.1MB

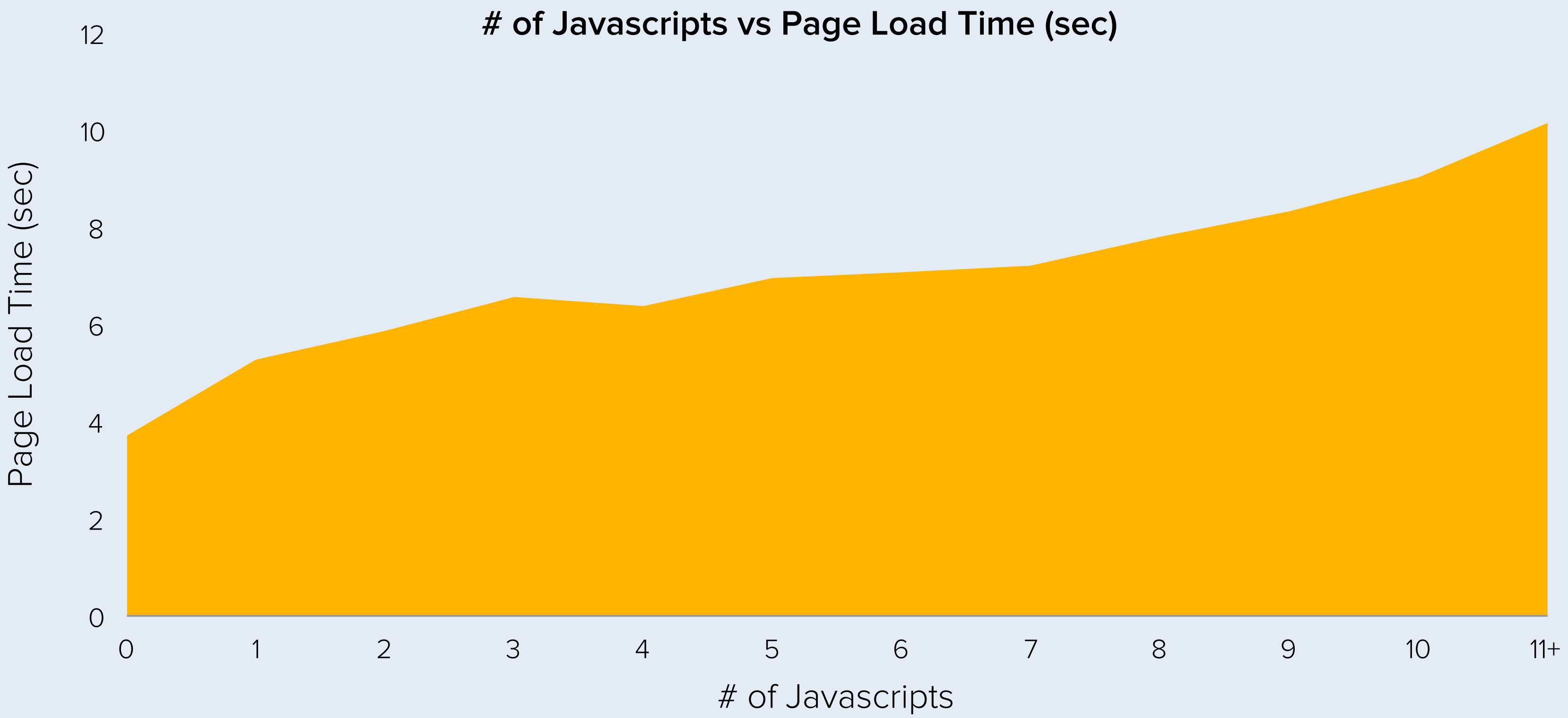
AVG PAGE SIZE

AVG SIZE OF TOTAL IMAGES	**653 KB**
AVG SIZE OF TOTAL JAVASCRIPT FILES	**173 KB**
AVG SIZE OF TOTAL CSS FILES	**34 KB**
AVG SIZE OF TOTAL OTHER FILES	**260 KB**


Page Size (KB) vs Page Load Time (sec)

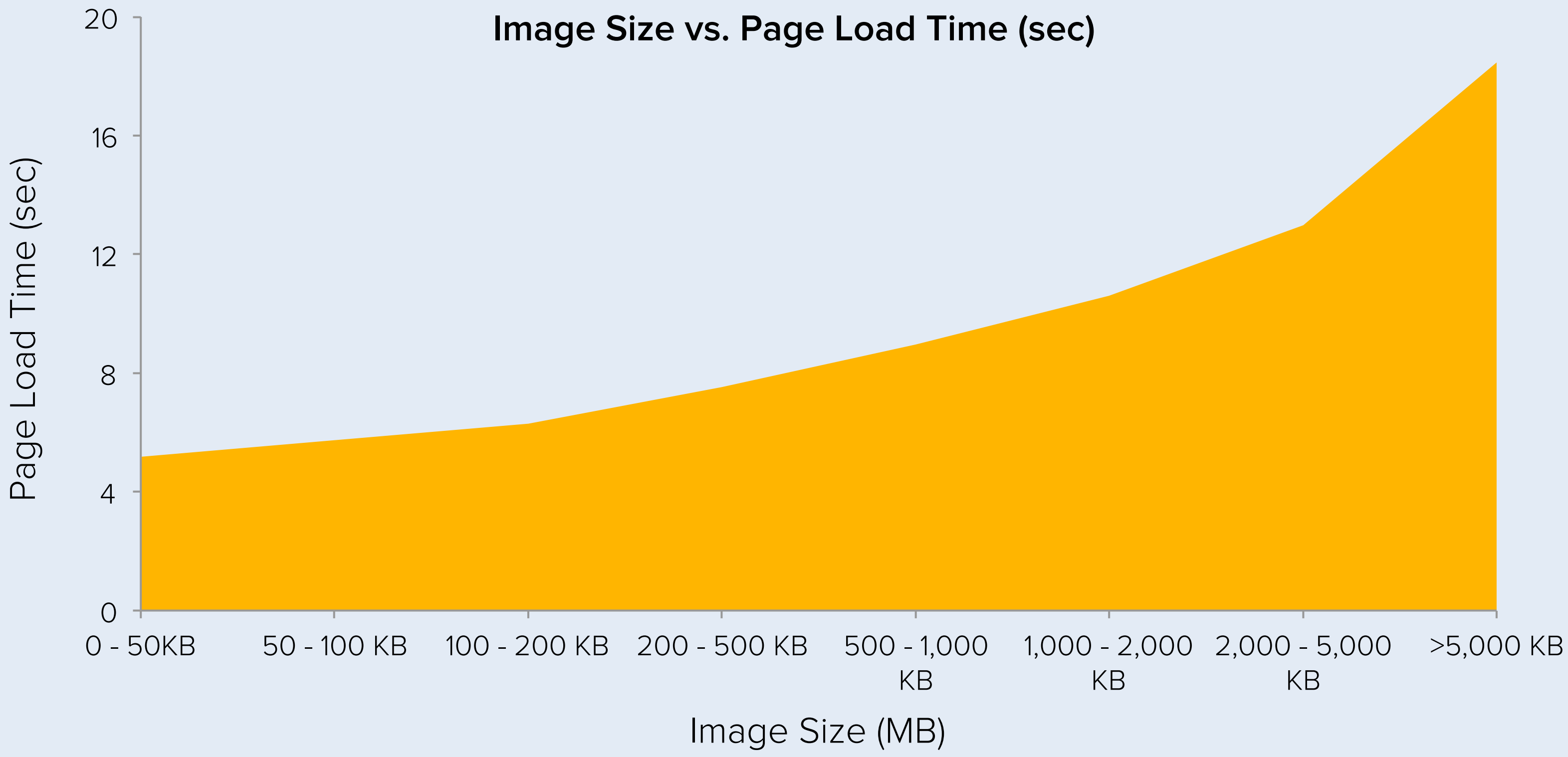
JAVASCRIPT BREAKDOWN

JavaScript is generally the worst offender of poor performance, especially as it relates to the number of JavaScript files.



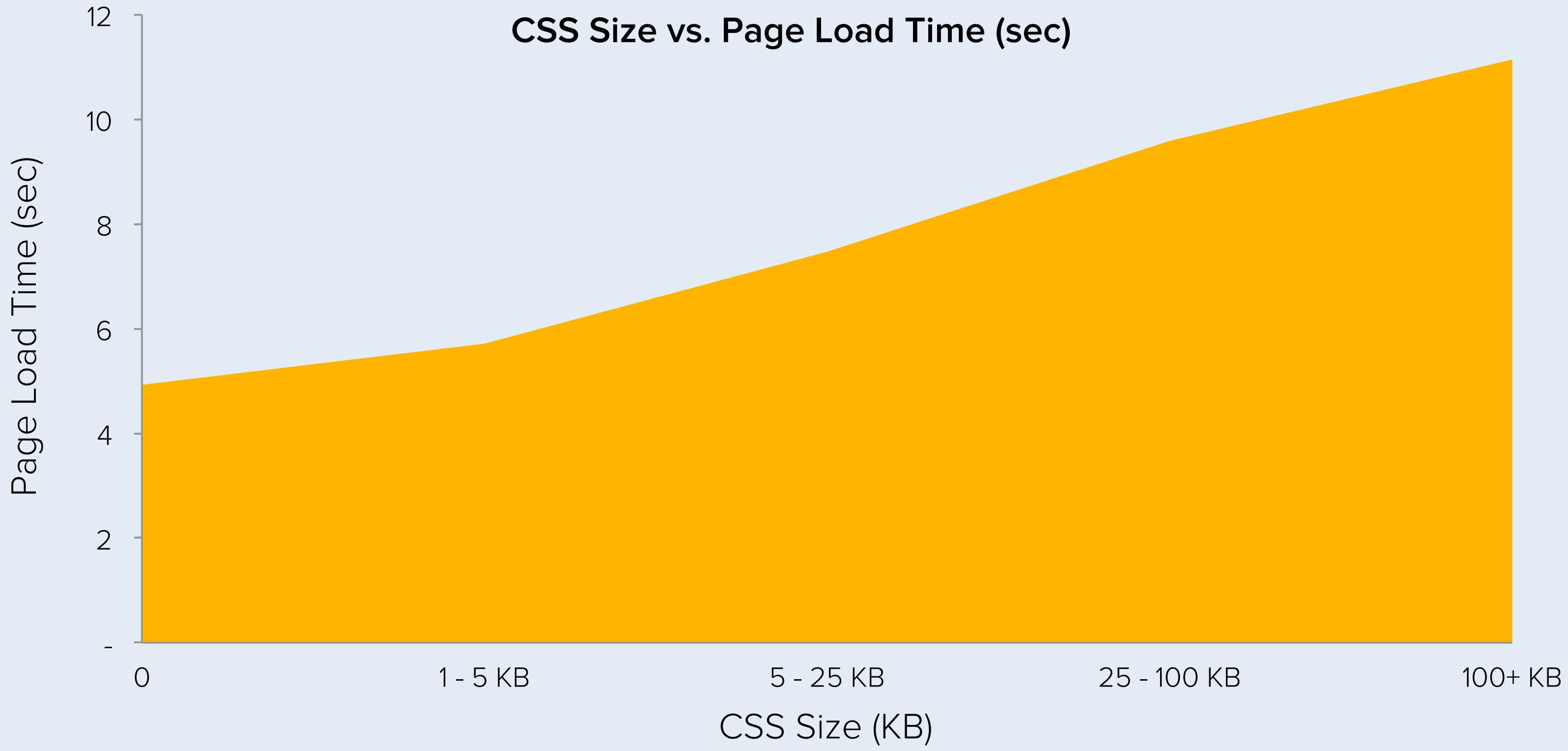
IMAGES BREAKDOWN

A high amount of images and large image sizes are also one of the top culprits when it comes to slowing down page loads.



CSS BREAKDOWN

CSS can be a contributing factor of poor performance, especially in terms of its size. Since CSS paints the visual display of a webpage, larger CSS files have a higher amount of display work needed to executed by the browser.



Chapter 4

Understanding Key Web PERFORMANCE METRICS

There are 3 key metric areas for web performance: back-end performance, front-end performance, and content complexity.

Each key area contains several metrics that will provide actionable insight to improve the web performance and page load times of websites.

Chapter 5

How to Optimize Your **WEBSITE ASSETS**

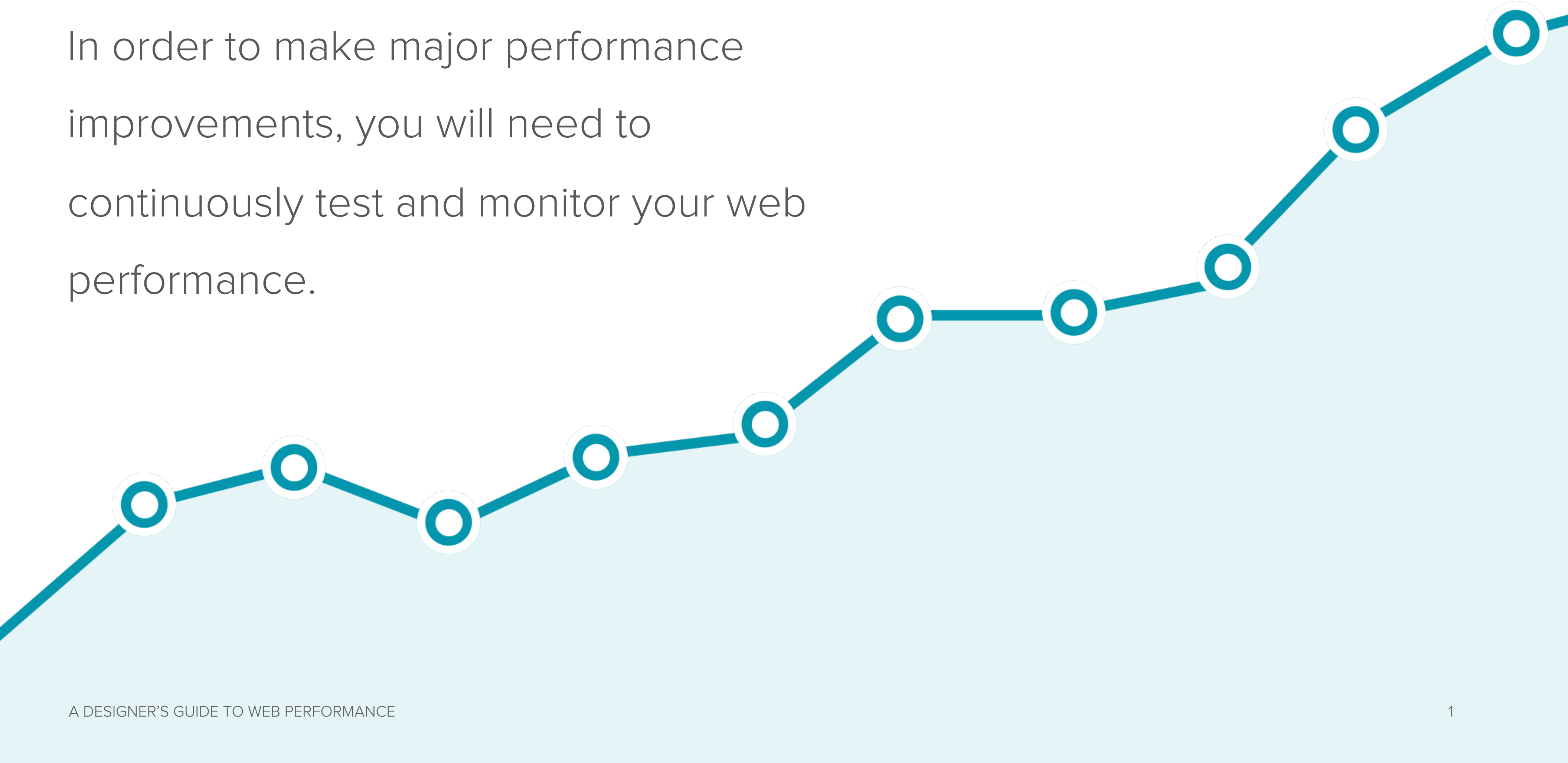
There are many key methods and tactics you can use to optimize your web performance.

Let’s take a deep dive into the best practices to optimize the individual assets of HTML, CSS, JavaScript, and images.

Chapter 6

How to Monitor Your **WEB PERFORMANCE**

In order to make major performance improvements, you will need to continuously test and monitor your web performance.



> If you can’t measure it, you can’t *improve* it.
>
> ***Lord Kelvin****, Physicist and Engineer*

The first thing you should do is test your website to establish a baseline of your current performance. You need to understand how your website and its assets load and where there are areas to improve. There are several key factors you should test your website against: different browsers, locations, and connectivities.

Testing your website only provides a single snapshot in time, however. To capture a holistic view, you'll need to monitor and track your performance over time. This way, you can understand your performance trends -- how your website performs across the world, with various browsers, different connectivities, and traffic spikes.

Chapter 7

Design With Performance AS A PRIORITY

The ultimate goal of this eBook is to create a fundamental shift in how you approach design: to design with performance as a priority.

By looking at web design through the lens of performance, you'll be able to build better, faster, and more scalable websites and web apps.