

Page Speed

Page speed is a measurement of how fast the content on your page loads.

What is Page Speed?

Page speed is often confused with "site speed," which is actually the page speed for a sample of page views on a site. Page speed can be described in either "page load time" (the time it takes to fully display the content on a specific page) or "time to first byte" (how long it takes for your browser to receive the first byte of information from the web server).

You can evaluate your page speed with [Google's PageSpeed Insights](#). PageSpeed Insights Speed Score incorporates data from CrUX (Chrome User Experience Report) and reports on two important speed metrics: First Contentful Paint (FCP) and DOMContentLoaded (DCL).

SEO Best Practices

Google has indicated site speed (and as a result, page speed) is [one of the signals used by its algorithm to rank pages](#). And [research has shown](#) that Google might be specifically measuring time to first byte as when it considers page speed. In addition, a slow page speed means that search engines can crawl fewer pages using their allocated crawl budget, and this could negatively affect your indexation.

Page speed is also important to user experience. Pages with a longer load time tend to have higher bounce rates and lower average time on page. Longer load times [have also been shown](#) to negatively affect conversions.

Here are some of the many ways to increase your page speed:

Enable compression

Use [Gzip](#), a software application for [file compression](#), to reduce the size of your CSS, HTML, and JavaScript files that are larger than 150 bytes.

Do not use gzip on image files. Instead, compress these in a program like Photoshop where you can retain control over the quality of the image. See "Optimize images" below.

Minify CSS, JavaScript, and HTML

By optimizing your code (including removing spaces, commas, and other unnecessary characters), you can dramatically increase your page speed. Also remove code comments, formatting, and unused code. [Google recommends using CSSNano and UglifyJS](#).

Reduce redirects

Each time a page redirects to another page, your visitor faces additional time waiting for the HTTP request-response cycle to complete. For example, if your mobile redirect pattern looks like this: "example.com -> [www.example.com](#) -> m.example.com -> m.example.com/home," each of those two additional redirects makes your page load slower.

Remove render-blocking JavaScript

Browsers have to build a DOM tree by parsing HTML before they can render a page. If your browser encounters a script during this process, it has to stop and execute it before it can continue.

[Google suggests avoiding and minimizing the use of blocking JavaScript.](#)

Leverage browser caching

Browsers cache a lot of information (stylesheets, images, JavaScript files, and more) so that when a visitor comes back to your site, the browser doesn't have to reload the entire page. Use a tool like [YSlow](#) to see if you already have an expiration date set for your cache. Then [set your "expires" header](#) for how long you want that information to be cached. In many cases, unless your site design changes frequently, a year is a reasonable time period. Google has more information about leveraging caching [here](#).

Improve server response time

Your server response time is affected by the amount of traffic you receive, the resources each page uses, the software your server uses, and the hosting solution you use. To improve your server response time, look for performance bottlenecks like slow database queries, slow routing, or a lack of adequate memory and fix them. The optimal server response time is under 200ms. Learn more about [optimizing your time to first byte](#).

Use a content distribution network

Content distribution networks (CDNs), also called content delivery networks, are networks of servers that are used to distribute the load of delivering content. Essentially, copies of your site are stored at multiple, geographically diverse data centers so that users have faster and more reliable access to your site.

Optimize images

Be sure that your images are no larger than they need to be, that they are in the right file format (PNGs are generally better for graphics with fewer than 16 colors while JPEGs are generally better for photographs) and that they are compressed for the web.

Use CSS sprites to create a template for images that you use frequently on your site like buttons and icons. CSS sprites combine your images into one large image that loads all at once (which means fewer HTTP requests) and then display only the sections that you want to show. This means that you are saving load time by not making users wait for multiple images to load.