





# Web performance resources

There are many reasons why your website should perform as well as possible.

Below is a quick review of best practices, tools, APIs with links to provide more information about each topic.

## Best practices

- Start with learning the [critical rendering path](#) of the browser. Knowing this will help you understand how to improve the performance of the site.
- Using *resource hints* such as [rel=preconnect](#), [rel=dns-prefetch](#), [rel=prefetch](#), [rel=preload](#).
- Keep the size of Javascript to a [minimum](#) . Only use as much Javascript as needed for the current page.
- [CSS](#) performance factors
- Use [HTTP/2](#) on your server (or CDN).
- Use a CDN for resources which can reduce load times significantly.
- Compress your resources using [gzip](#) , [Brotli](#) , and [Zopfli](#) .
- Image optimization (use CSS animation, or SVG if possible).
- Lazy loading parts of your application outside the viewport. If you do, have a backup plan for SEO (e.g render full page for bot traffic); for example, by using the [loading](#) attribute on the [<img>](#) element
- It is also crucial to realize what is really important to your users. It might not be absolute timing, but [user perception](#).

## Quick Wins

### CSS

Web performance is all about user experience and perceived performance. As we learned in the [critical rendering path](#) document, linking CSS with a traditional link tag with `rel="stylesheet"` is synchronous and blocks rendering. Optimize the rendering of your page by removing blocking CSS.

To load CSS asynchronously one can set the media type to print and then change to all once loaded. The following snippet includes an onload attribute, requiring Javascript, so it is important to include a noscript tag with a traditional fallback.

```
<link rel="stylesheet" href="/path/to/my.css" media="print" onload="this.media='all'">
<noscript><link rel="stylesheet" href="/path/to/my.css"></noscript>
```



The downside with this approach is the flash of unstyled text (FOUT.) The simplest way to address this is by inlining CSS that is required for any content that is rendered above the fold, or what you see in the browser viewport before scrolling. These styles will improve perceived performance as the CSS does not require a file request.

```
<style type="text/css">
// Insert your CSS here
</style>
```



### Javascript

Avoid Javascript blocking by using the [async](#) or [defer](#) attributes, or link javascript assets after the page's DOM elements. Javascript only block rendering for elements that appear after the script tag in the DOM tree.

### Web Fonts

EOT and TTF formats are not compressed by default. Apply compression such as GZIP or Brotli for these file types. Use WOFF and WOFF2. These formats have compression built in.

Within `@font-face` use `font-display: swap`. By using font display swap the browser will not block rendering and will use the backup system fonts that are defined. Optimize [font weight](#) to match the web font as

closely as possible.

## Icon web fonts

If possible avoid icon web fonts and use compressed SVGs. To further optimize inline your SVG data within HTML markup to avoid HTTP requests.

# Tools

- Learn to use the [Firefox Dev Tools](#) to profile your site.
- [PageSpeed Insights](#) [↗](#) can analyze your page and give some general hints to improve performance.
- [Lighthouse](#) [↗](#) can give you a detailed breakdown of many aspects of your site including performance, SEO and accessibility.
- Test your page's speed using [WebPageTest.org](#) [↗](#), where you can use different real device types and locations.
- Try the [Chrome User Experience Report](#) [↗](#) which quantifies real user metrics.
- Define a [performance budget](#).

## APIs

- Gather user metrics using <https://github.com/akamai/boomerang> [↗](#)
- Or directly gather with [window.performance.timing](#)

## Things not to do (bad practices)

- Download everything
- Use uncompressed media files