Choose the right image format

- Choose the right image format
- Implications of high-resolution screens
- Features of different raster image formats
- Impact on Largest Contentful Paint (LCP)

The very first question you should ask yourself is whether an image is, in fact, required to achieve the effect you are after. Good design is simple and will also always yield the best performance. If you can eliminate an image resource, which often requires a large number of bytes relative to HTML, CSS, JavaScript and other assets on the page, then that is always the best optimization strategy. That said, a well-placed image can also communicate more information than a thousand words, so it is up to you to find that balance.

Next, you should consider if there is an alternative technology that could deliver the desired results, but in a more efficient manner:

- **CSS effects** (such as shadows or gradients) and CSS animations can be used to produce resolution-independent assets that always look sharp at every resolution and zoom level, often at a fraction of the bytes required by an image file.
- **Web fonts** enable use of beautiful typefaces while preserving the ability to select, search, and resize text—a significant improvement in usability.

If you ever find yourself encoding text in an image asset, stop and reconsider. Great typography is critical to good design, branding, and readability, but text-in-images delivers a poor user experience: the text is not selectable, not searchable, not zoomable, not accessible, and not friendly for high-DPI devices. The use of web fonts requires its own set of optimizations, but it addresses all of these concerns and is always a better choice for displaying text.

Choose the right image format

If you are sure an image is the correct option, you should carefully select the right kind of image for the job.

Zoomed-in vector and raster images

Zoomed-in vector image (L) raster image (R)

- Vector graphics use lines, points, and polygons to represent an image.
- Raster graphics represent an image by encoding the individual values of each pixel within a rectangular grid.

Each format has its own set of pros and cons. Vector formats are ideally suited for images that consist of simple geometric shapes such as logos, text, or icons. They deliver sharp results at every resolution and zoom setting, which makes them an ideal format for high-resolution screens and assets that need to be displayed at varying sizes.

However, vector formats fall short when the scene is complicated (for example, a photo): the amount of SVG markup to describe all the shapes can be prohibitively high and the output may still not look "photorealistic". When that's the case, that's when you should be using a raster image format such as PNG, JPEG, WebP, or AVIF.

Raster images do not have the same nice properties of being resolution or zoom independent —when you scale up a raster image you'll see jagged and blurry graphics. As a result, you may need to save multiple versions of a raster image at various resolutions to deliver the optimal experience to your users.

Implications of high-resolution screens

There are two different kinds of pixels: CSS pixels and device pixels. A single CSS pixel may correspond directly to a single device pixel, or may be backed by multiple device pixels. What's the point? Well, the more device pixels there are, the finer the detail of the displayed content on the screen.

Three images showing the difference between CSS pixels and device pixels.

The difference between CSS pixels and device pixels.

High DPI (HiDPI) screens produce beautiful results, but there is one obvious tradeoff: image assets require more detail to take advantage of the higher device pixel counts. The good news is, vector images are ideally suited for this task, as they can be rendered at any resolution with sharp results— you might incur a higher processing cost to render the finer detail, but the underlying

asset is the same and is resolution independent.

On the other hand, raster images pose a much larger challenge because they encode image data on a per-pixel basis. Hence, the larger the number of pixels, the larger the filesize of a raster image. As an example, let's consider the difference between a photo asset displayed at 100x100 (CSS) pixels:

Screen resolution	Total pixels	Uncompressed filesize (4 bytes per pixel)
1x	100 x 100 = 10,000	40,000 bytes
2x	100 x 100 x 4 = 40,000	160,000 bytes
3x	100 x 100 x 9 = 90,000	360,000 bytes

When we double the resolution of the physical screen, the total number of pixels increases by a factor of four: double the number of horizontal pixels, times double the number of vertical pixels. Hence, a "2x" screen not just doubles, but quadruples the number of required pixels!

So, what does this mean in practice? High-resolution screens enable you to deliver beautiful images, which can be a great product feature. However, high-resolution screens also require high-resolution images, therefore:

- Prefer vector images whenever possible as they are resolution-independent and always deliver sharp results.
- If a raster image is required, serve responsive images.

Features of different raster image formats

In addition to different lossy and lossless compression algorithms, different image formats support different features such as animation and transparency (alpha) channels. As a result, the choice of the "right format" for a particular image is a combination of desired visual results and functional requirements.

Format	Transparency	Animation	Browser
PNG	Yes	No	All
JPEG	No	No	All
WebP	Yes	Yes	All modern browsers. See Can I use?

Format	Transparency	Animation	Browser
AVIF	Yes	Yes	No. See Can I use?

There are two universally supported raster image formats: PNG and JPEG. In addition to these formats, modern browsers support the newer format WebP, while only some support the newer AVIF format. Both of the newer formats offer better overall compression and more features. So, which format should you use?

WebP and AVIF will generally provide better compression than older formats, and should be used where possible. You can use WebP or AVIF images along with a JPEG or PNG image as a fallback.

See Use WebP images for more details.

In terms of older image formats, consider the following:

1. **Do you need animation? Use `<video>` elements.**
   - What about GIF? GIF limits the color palette to at most 256 colors, and creates significantly larger file sizes than `<video>` elements. See Replace animated GIFs with video.
2. **Do you need to preserve fine detail with highest resolution? Use PNG or lossless WebP.**
   - PNG does not apply any lossy compression algorithms beyond the choice of the size of the color palette. As a result, it will produce the highest quality image, but at a cost of significantly higher filesize than other formats. Use judiciously.
   - WebP has a lossless encoding mode that may be more efficient than PNG.
   - If the image asset contains imagery composed of geometric shapes, consider converting it to a vector (SVG) format!
   - If the image asset contains text, stop and reconsider. Text in images is not selectable, searchable, or "zoomable". If you need to convey a custom look (for branding or other reasons), use a web font instead.
3. **Are you optimizing a photo, screenshot, or a similar image asset? Use JPEG, lossy WebP, or lossy AVIF.**
   - JPEG uses a combination of lossy and lossless optimization to reduce filesize of the image asset. Try several JPEG quality levels to find the best quality versus filesize tradeoff for your asset.
   - Lossy WebP or lossy AVIF may be acceptable JPEG alternatives, but be aware that WebP's lossy mode in particular discards some chroma information to achieve smaller images. This means that select colors may not be the same as an equivalent JPEG.

Finally, note that if you are using a WebView to render content in your platform-specific application, then you have full control of the client and can use WebP exclusively! Facebook and many others use WebP to deliver all of their images within their applications— the savings are definitely worth it.

Impact on Largest Contentful Paint (LCP)

Images may be LCP candidates. That means the size of an image affects its load time. When an image is an LCP candidate, efficiently encoding that image is crucial to improving LCP.

You should strive to apply the advice given in this article so that the perceptual performance of a page is as fast as it can possibly be for all users. LCP is part of perceptual performance, as it measures how fast the largest (and therefore most perceivable) element on the page displays.

□□□ #3
□ □□□ □□□ 2026-10-31 22:33:13 UTC
□ cocobook-sync □□□ 2026-11-04 17:38:43 UTC