WebP vs JPEG vs PNG: which image format to use, and why

Picking the wrong image format is the quiet reason a page is heavier than it needs to be. Here is a plain guide to JPEG, PNG, WebP and AVIF in 2026: what each one is good at, the one rule that decides most cases, and why resizing matters more than the format you choose.

Published 2026-06-30 · A Mue guide

Image format is one of those choices that looks trivial and quietly decides how fast your pages load. Pick well and a photo that was two megabytes ships as two hundred kilobytes with no visible difference. Pick badly and you serve a lossless screenshot format for a holiday photo and wonder why the page crawls. The good news is that the decision comes down to a handful of formats and one rule that settles most cases. Here is the plain version for 2026.

The four formats that matter

You will meet four image formats on the web, and each is built for a different job:

- JPEG is lossy and built for photographs. It throws away detail the eye barely notices to get small files, which is exactly right for photos and exactly wrong for sharp text or flat graphics, where it leaves visible artefacts. It has no transparency and no animation, but it opens literally everywhere, which is its enduring advantage.
- PNG is lossless and built for graphics. It keeps every pixel exactly, supports true transparency through an alpha channel, and is the right pick for logos, icons, screenshots and line art. Used on a photograph it produces a file several times larger than it should be.
- WebP is the modern all-rounder. It has both a lossy mode (smaller than JPEG) and a lossless mode (smaller than PNG), plus transparency and animation. It is the format that can replace either of the two above in most cases.
- AVIF is the newest and the most efficient, usually smaller again than WebP at the same quality, with strong support for HDR and wide colour. It encodes more slowly and is not produced by every tool yet, but it is the direction the web is heading.

The one rule that decides most cases

Before the format names blur together, anchor on the distinction that settles the majority of choices: is the image a photograph or a graphic?

- Photographs, meaning anything with smooth gradients and no hard edges, compress beautifully with lossy formats. Reach for WebP first, with JPEG as the universal fallback, and AVIF if your pipeline produces it.
- Graphics, meaning logos, icons, screenshots with text, charts and line art, need crisp edges and often transparency. Reach for PNG, or lossless WebP for the same result at a smaller size.

Almost every real decision is a variation on that split. A product photo is a photograph; your logo on top of it is a graphic. When an image is genuinely both, such as a screenshot that contains a photo, favour the dominant content and check the result at full size.

How much WebP actually saves

The case for WebP is not marketing, it is measured. Google's published tests put lossy WebP roughly 25 to 34 percent smaller than a JPEG of the same visual quality, and lossless WebP about 26 percent smaller than an equivalent PNG. On a page with a dozen images that is the difference between a fast load and a slow one, especially on mobile connections. AVIF typically goes further still, which is why the modern pattern is to serve AVIF where a browser accepts it, fall back to WebP, and keep a JPEG as the last resort for anything that has to open outside a browser.

Support is no longer the obstacle it once was. Every current major browser reads WebP, and Safari has done since version 14 in 2020. AVIF is read by Chrome, Firefox and Safari 16 and later. The remaining rough edges are off the web, in some older email clients and legacy desktop software, which is the only reason a JPEG fallback still earns its place.

The saving most people miss: resize before you convert

Here is the part that matters more than any format argument. The biggest reason web images are too large is not the format, it is the dimensions. A photo from a phone is often 4000 pixels wide, and a layout almost never shows it larger than about 1200 to 2000 pixels. Converting that image to WebP without resizing still ships a picture several times bigger than the slot it fills. Downscaling to the size it will actually display is the single largest saving available, and it comes before format choice, not after.

Two smaller levers finish the job. Dropping lossy quality to around 75 to 85 percent removes detail the eye does not register on a typical screen, and stripping the embedded camera metadata (the EXIF block, which can also carry the GPS location a photo was taken at) trims a little more while removing a small privacy leak. Do those three things, resize, sensible quality, strip metadata, and the format you land on matters far less than it seemed.

A quick decision list

- Photo for the web: WebP at 75 to 85 percent quality, resized to its display width. Keep a JPEG fallback for anything that leaves your site.
- Logo, icon, screenshot or anything with text or transparency: PNG, or lossless WebP for a smaller file.
- You need the smallest possible files and your tools support it: AVIF, with a WebP fallback behind it.
- A picture from an iPhone in HEIC: convert it to WebP or JPEG for the web, since HEIC is not widely readable in browsers.
- An old animated GIF: re-encode it to WebP, which carries animation at a fraction of the size.

None of this needs an account or an upload. The image converter on this site runs entirely in

your browser, converts between WebP, JPG and PNG, and lets you resize and set quality in the same step, so the file never leaves your device. Pick the format from the rule above, resize to the size you actually need, and most of the weight takes care of itself.

Frequently asked questions

Is WebP better than JPEG?

For most web photos, yes. WebP supports the same lossy compression JPEG does but is more efficient at it: Google's own tests put lossy WebP roughly 25 to 34 percent smaller than a JPEG of equivalent visual quality. WebP also does two things JPEG cannot, namely transparency and animation, and it has a lossless mode for graphics. The one reason to still reach for JPEG is reach: every device and every old email client opens a JPEG, so it remains the safest universal fallback. For a modern website, WebP is the better default and JPEG is the fallback.

When should I still use PNG?

Use PNG when you need crisp edges or transparency: logos, icons, screenshots with text, line art, and anything layered over another colour. PNG is lossless, so it keeps hard edges sharp where JPEG would smear them with compression artefacts, and it carries an alpha channel for true transparency. The catch is that PNG is a poor choice for photographs, where it produces files several times larger than JPEG or WebP for no visible benefit. If you want PNG-style sharpness and transparency at a smaller size, lossless WebP does the same job and is usually smaller.

Is WebP supported everywhere now?

For practical purposes, yes. Every current major browser has read WebP for years: Chrome and Firefox for a long time, Safari since version 14 in 2020, and Edge. The places it can still trip you up are outside the browser, such as some older email clients and a few legacy desktop apps, which is why a JPEG fallback is still sensible for anything that leaves your site. On the open web in 2026, serving WebP is a safe default.

Should I use AVIF instead of WebP?

AVIF compresses even better than WebP, often noticeably smaller again at the same quality, and it handles HDR and wide colour well. Browser support is now broad, with Chrome, Firefox and Safari 16 and later all reading it. The trade-offs are that AVIF encodes more slowly and not every tool exports it yet. A common pattern is to serve AVIF with a WebP fallback and a JPEG behind that. If your workflow does not produce AVIF, you are not missing much by shipping WebP, which is why this converter outputs WebP, JPG and PNG.

Why is my converted image still large?

Usually because the dimensions, not the format, are the problem. A photo straight off a phone can be 4000 pixels wide, and converting it to WebP without resizing still leaves a far bigger image than a web layout needs. The single biggest saving is to downscale to the size it will actually display, often 1200 to 2000 pixels wide at most, before or during conversion. After that, dropping lossy quality to around 75 to 85 percent and stripping the embedded camera metadata trims the rest. Format choice helps, but resizing is what turns a multi-megabyte file into a small one.

Convert and compress images in your browser
Free, no signup: agent.mue.app/tools/image-converter

agent.mue.app/articles/webp-vs-jpeg-vs-png-which-image-format