Engineer

WebP

a raster image format with lossy and lossless compression

What it is

WebP is a raster image format published by Google, stored in files with the extension `.webp` and served with the registered media type `image/webp`. One format covers jobs that used to take three: photographs, which went out as JPEG; flat graphics with hard edges and transparency, which went out as PNG; and short loops, which went out as GIF. In prose it is written with a capital W and a capital P, and on disk it is simply the `.webp` file.

Inside, a WebP file is a RIFF container, the chunked layout that WAV audio also uses. It supports two image bitstreams. A `VP8` chunk carries lossy data, encoded with the key-frame method of the VP8 video codec: the encoder predicts each block of pixels from neighbours it has already decoded and stores only the difference. A `VP8L` chunk carries lossless data, from a separate coder written for pictures that must come back bit for bit. A `VP8X` header marks the extended form, which can add an alpha channel, animation frames, an ICC colour profile and EXIF or XMP metadata. Google documented the format alongside its own library, and it has since been published as RFC 9649, an informational RFC from November 2024.

The extended form is what makes the format more than a smaller JPEG. A lossy picture may carry a separate alpha channel, so a photograph with a soft, transparent edge no longer has to be a heavy lossless file. Animation sits in the same container, frame by frame, each frame lossy or lossless as suits it.

The image bitstreams have their own limits. Each side of an encoded picture or frame is stored in 14 bits, so it tops out at 16,383 pixels for lossy data and 16,384 for lossless. The extended form can place frames on a larger canvas, and a whole file stays just under 4 GiB. Every channel is stored in 8 bits, which suits pictures meant for a screen and loses the extra depth of a 16-bit master.

Use the concept in a decision

Put “What is WebP?” to work in a real decision. We can help relate the concept to the system you are building or buying. Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Directed full-stack GIS map development — full-stack mapping, from database to browser

Directed full-stack GIS map development Get in touch →

Why it matters

Pictures are usually the heaviest bytes on a page, and the format decides how many of them there are. Google’s own comparison puts lossless WebP 26% below PNG, and lossy WebP 25–34% below JPEG at equal SSIM, a measure of structural similarity between two images. A lossy WebP with an alpha channel typically comes to a third of the PNG it replaces. Fewer bytes reach the screen sooner on a slow connection, cost less to serve, and leave room in a page’s budget for the things the reader came for.

The decision a team faces is less whether to use WebP than which format to ship for which picture, and whether to ship more than one. Published averages stop helping at that point. They are averages over a test corpus, while the files on a real site are a logo, a handful of photographs and some screenshots, and the ranking can come out differently for each. AVIF, the newer format, is the usual rival, and on paper it compresses harder. On a given site, at the sizes that site renders, it may lose.

The second decision is where conversion happens. Converting by hand once leaves a folder of files that drift away from their sources. Converting at build time, from an original kept in the repository, makes every published size reproducible and makes it cheap to change the quality setting, or the format, later.

The third is everything outside the browser. A picture’s address travels: into feeds, e-mail, chat previews, office documents and image editors, and each of those decides for itself which formats it reads. A site that moves every picture to WebP learns which receivers accept it from the first reader who sees a broken image, unless it checks each path first.

What it looks like in practice

The credits page lists this site's pictures as WebP and PNG images, sized at build time. The templates show what that means. Inline photographs, avatars and the brand mark all pass through one shared partial that resizes the original into several widths, each encoded as WebP, and writes a single `img` element with a `srcset`, so the browser picks the width and the site has already picked the format.

That format is a measurement rather than a default. The site's Hugo version can encode AVIF, so serving both through a `picture` element was an option. In the comparison documented on 17 August 2026, at the sizes the site rendered, AVIF came out 30–36% larger than WebP for the brain mark and 3–7% larger for the avatars; it won only on the portrait, by 13%, and barely on the office photographs, by 3–4%. A `picture` element chooses its source by order and by what the browser supports, never by size, so listing AVIF first would have sent every modern browser about a third more bytes for the logo on every page. With one format worth shipping, the template uses `img` with `srcset` and no `picture` at all.

The same partial carries a second lesson. Hugo does not pass an image through when asked to resize it to the width it already has: it re-encodes it, making a second published file and, for a lossy original, another lossy pass. When a requested width equals the original's, the partial therefore names the original directly by default. Callers that explicitly ask for every size to be derived still re-encode it; the photo shortcode takes that path to avoid publishing a second copy of a master that is already served by the assets mount.

WebP stops at the edge of what other software reads. The RSS feed points its `image` element at a PNG, because RSS 2.0 accepts only GIF, JPEG and PNG there and caps the picture at 144 by 400 pixels: the 1,280-pixel WebP share card drew a warning from the W3C feed validator, and the 144-pixel apple-touch icon, a PNG at exactly the maximum width, took its place. The screenshots that the English and Danish manifests hand to Chromium's install dialog, on the other hand, are WebP, a wide capture and a narrow one for each, and a lint task fails when their recorded generation stamp trails a committed change to the rendering inputs it tracks; uncommitted edits are outside that check.

Where it goes wrong

A converted JPEG loses detail twice. Turning an existing JPEG into lossy WebP decodes one lossy generation and encodes another, and the artefacts of both survive. Hugo's resize-to-the-same-width behaviour is a quieter form of the same trap. Keep a lossless or full-quality original in the repository and encode every published file from that.

Colours turn dull after conversion. `cwebp`, the reference encoder, copies no metadata unless told to, so the ICC profile of a wide-gamut photograph is left behind and the browser treats the pixels as sRGB. Pass `-metadata icc`, or convert the picture to sRGB before encoding.

A feed reader shows a broken picture. Software written before the format, or held to an older specification, treats a WebP as an unknown file. RSS 2.0 is the clearest case: its `image` element names three formats, and WebP is outside them. Follow each place a picture's address travels, and serve PNG or JPEG where the receiver demands it.

A long screenshot stops the encoder. A full-page capture of a long article easily passes 16,383 pixels in height. The libwebp encoder caps each side at that size, even though the lossless bitstream can represent one more pixel. Split the capture, or scale it down first.

The bigger file wins the `picture` element. Listing a newer format first sends it to every browser that decodes it, whether or not it is smaller for that image. Measure each candidate at the sizes the page renders before adding a source, as this site did with AVIF.

Further reading

- WebP — Google's home for the format: the compression study behind the published percentages, the libwebp library and its tools.
- RFC 9649: WebP Image Format — the container, the lossy and lossless bitstreams, extended features and size limits, as an informational RFC.
- cwebp — every option of the reference encoder, including `-q`, `-lossless` and `-metadata`.
- Hugo: Image processing — how a Hugo site resizes and converts images, WebP included, at build time.

- HTML Standard: Images — how `srcset`, `sizes` and `picture` let the browser choose among candidate files.
- RSS 2.0 Specification — the `image` element, the three formats it accepts and its size limits.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is WebP?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/webp/