Engineer

# WebP

et rasterbilledformat med komprimering med og uden tab

## Hvad det er

WebP er et rasterformat til billeder, udgivet af Google, gemt i filer med endelsen `.webp` og serveret med den registrerede medietype `image/webp`. Ét format dækker opgaver, der før krævede tre: fotografier, som blev sendt som JPEG; flad grafik med skarpe kanter og gennemsigtighed, som blev sendt som PNG; og korte animerede løkker, som blev sendt som GIF. Navnet skrives med stort W og stort P, og på disken er det blot `.webp` -filen.

Indvendigt er en WebP-fil en RIFF-beholder, den opdelte struktur, som WAV-lydfiler også bruger. Den understøtter to billedbitstrømme. En `VP8` -blok bærer data med tab, kodet med nøglebilledmetoden fra videocodecen VP8: koderen forudsiger hver blok af pixel ud fra naboer, den allerede har afkodet, og gemmer kun forskellen. En `VP8L` -blok bærer tabsfri data fra en separat koder, skrevet til billeder, der skal komme tilbage bit for bit. En `VP8X` -header markerer den udvidede form, som kan tilføje en alfakanal, animationsbilleder, en ICC-farveprofil og metadata i EXIF eller XMP. Google dokumenterede formatet sammen med sit eget bibliotek, og siden er det udgivet som RFC 9649, en informativ RFC fra november 2024.

Det er den udvidede form, der gør formatet til mere end en mindre JPEG. Et billede med tab kan have en separat alfakanal, så et fotografi med en blød, gennemsigtig kant slipper for at være en tung tabsfri fil. Animation ligger i samme beholder, billede for billede, og hvert billede kodes med eller uden tab, alt efter hvad der passer.

Billedbitstrømmene har deres egne grænser. Hver side af et kodet billede eller animationsbillede gemmes i 14 bit, så den højst kan være 16.383 pixel for data med tab og 16.384 for tabsfri data. Den udvidede form kan placere billeder på et

større lærred, og en hel fil holder sig lige under 4 GiB. Hver kanal gemmes i 8 bit, hvilket passer til billeder beregnet til en skærm og taber den ekstra dybde i en original på 16 bit.

Brug begrebet i en beslutning

Brug »Hvad er WebP?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Brugerflader, der indlæses hurtigt, virker på en telefon og opfører sig ens i alle de browsere, jeres kunder nu bruger.

**Se vores ydelser** Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har styret full-stack-udviklingen af GIS-kort — full stack-kortløsning, fra database til browser

**Har styret full-stack-udviklingen af GIS-kort** Kontakt os →

## Hvorfor det betyder noget

Billeder er som regel de tungeste bytes på en side, og formatet afgør, hvor mange af dem der er. Googles egen sammenligning placerer tabsfri WebP 26 % under PNG og WebP med tab 25–34 % under JPEG ved samme SSIM, et mål for strukturel lighed mellem to billeder. En WebP med tab og alfakanal fylder typisk en tredjedel af den PNG, den erstatter. Færre bytes når hurtigere frem på en langsom forbindelse, koster mindre at levere og efterlader plads i sidens budget til det, læseren kom efter.

Den beslutning, et team står over for, handler mindre om, hvorvidt det skal bruge WebP, end om hvilket format der skal udgives for hvilket billede, og om der skal udgives mere end ét. Dér holder de offentliggjorte gennemsnit op med at hjælpe. De er gennemsnit over et testkorpus, mens filerne på et rigtigt site er et logo, en håndfuld fotografier og nogle skærmbilleder, og rækkefølgen kan falde forskelligt ud for hver af dem. AVIF, det nyere format, er den sædvanlige rival, og på papiret komprimerer det hårdere. På et bestemt site, ved de størrelser det site viser, kan det tabe.

Den anden beslutning er, hvor konverteringen sker. Konverterer man i hånden én gang, står man tilbage med en mappe filer, der glider væk fra deres kilder. Konvertering under buildet, ud fra en original i repositoriet, gør hver udgivet størrelse reproducerbar og gør det billigt at ændre kvalitetsindstillingen, eller formatet, senere.

Den tredje er alt uden for browseren. Et billedes adresse rejser: ind i feeds, e-mail, forhåndsvisninger i chat, kontordokumenter og billedredigeringsprogrammer, og hver af dem afgør selv, hvilke formater den læser. Et site, der flytter alle billeder til WebP, lærer, hvilke modtagere der accepterer det, af den første læser, der ser et ødelagt billede, medmindre det kontrollerer hver vej på forhånd.

## Sådan ser det ud i praksis

Kolofonen opgiver dette websteds billeder som WebP- og PNG-billeder, skaleret under buildet. Skabelonerne viser, hvad det betyder. Fotografier i sideindholdet, avatarer og brandmærket går alle gennem én fælles partial, der skalerer originalen til flere bredder, hver kodet som WebP, og skriver ét `img` -element med en `srcset` , så browseren vælger bredden, og sitet allerede har valgt formatet.

Det format er en måling og ikke en standardindstilling. Sitets Hugo-version kan kode AVIF, så det var en mulighed at levere begge gennem et `picture` -element. I sammenligningen, der blev dokumenteret 17. august 2026, var AVIF ved de størrelser, sitet viste, 30–36 % større end WebP for hjernemærket og 3–7 % større for avatarerne; det vandt kun på portrættet, med 13 %, og knap nok på kontorfotografierne, med 3–4 %. Et `picture` -element vælger sin kilde efter rækkefølge og efter, hvad browseren understøtter, aldrig efter størrelse, så hvis AVIF stod først, ville hver moderne browser have fået omtrent en tredjedel flere bytes for logoet på hver side. Med ét format, der er værd at udgive, bruger skabelonen `img` med `srcset` og slet intet `picture` .

Den samme partial rummer en lektie mere. Hugo lader ikke et billede slippe uændret igennem, når det bliver bedt om at skalere det til den bredde, det allerede har: det koder billedet om og laver en ekstra udgivet fil og, for en original med tab, endnu en omgang med tab. Når en ønsket bredde er lig originalens, peger partialen derfor som standard direkte på originalen. Kaldere, der udtrykkeligt beder om, at hver størrelse afledes, koder det stadig om; photo-shortcoden vælger den vej for ikke at udgive endnu en kopi af en master, der allerede leveres gennem assets-mountet.

WebP holder op ved grænsen for, hvad anden software læser. RSS-feedet peger sit `image` -element på en PNG, fordi RSS 2.0 kun accepterer GIF, JPEG og PNG dér og begrænser billedet til 144 gange 400 pixel: delingskortet i WebP, 1.280 pixel bredt, fik en advarsel fra W3C's feedvalidator, og apple-touch-ikonet på 144

pixel, en PNG i præcis den største tilladte bredde, tog dets plads. De skærmbilleder, som det engelske og det danske manifest giver til Chromiums installationsdialog, er derimod WebP, et bredt og et smalt til hvert, og en lint-opgave fejler, når deres registrerede genereringstidspunkt ligger før en committet ændring i de renderingsinput, kontrollen følger; ændringer, der ikke er committet, ligger uden for den kontrol.

## Hvor det går galt

**En konverteret JPEG mister detaljer to gange.** Når en eksisterende JPEG laves om til WebP med tab, afkodes én generation med tab, og en ny kodes, og artefakterne fra begge overlever. Hugos omkodning ved samme bredde er en stillere udgave af den samme fælde. Hold en tabsfri original eller en original i fuld kvalitet i repositoriet, og kod hver udgivet fil ud fra den.

**Farverne bliver matte efter konverteringen.** `cwebp`, referencekoderen, kopierer ingen metadata, medmindre den bliver bedt om det, så ICC-profilen fra et fotografi med bred farveskala bliver efterladt, og browseren behandler billedpunkterne som sRGB. Angiv `-metadata icc`, eller konverter billedet til sRGB før kodningen.

**En feedlæser viser et ødelagt billede.** Software skrevet før formatet, eller bundet til en ældre specifikation, behandler en WebP som en ukendt fil. RSS 2.0 er det tydeligste tilfælde: dets `image`-element nævner tre formater, og WebP står uden for dem. Følg hvert sted, et billedes adresse rejser hen, og lever PNG eller JPEG, hvor modtageren kræver det.

**Et langt skærmbillede standser koderen.** En helsidesoptagelse af en lang artikel kommer let over 16.383 pixel i højden. Koderen i libwebp begrænser hver side til den størrelse, selv om den tabsfri bitstrøm kan rumme én pixel mere. Del optagelsen op, eller skaler den ned først.

**Den større fil vinder `picture`-elementet.** Står et nyere format først, sendes det til hver browser, der kan afkode det, uanset om det er mindre for netop det billede. Mål hver kandidat ved de størrelser, siden viser, før en kilde tilføjes, som dette site gjorde med AVIF.

# Læs mere

- WebP — Googles side for formatet: kompressionsundersøgelsen bag de offentliggjorte procenter, biblioteket libwebp og dets værktøjer.
- RFC 9649: WebP Image Format — beholderen, bitstrømmene med og uden tab, udvidede funktioner og størrelsesgrænser, udgivet som informativ RFC.
- cwebp — alle referencekoderens indstillinger, herunder `-q`, `-lossless` og `-metadata`.
- Hugo: Image processing — hvordan et Hugo-site skalerer og konverterer billeder, WebP inklusive, under buildet.
- HTML Standard: Images — hvordan `srcset`, `sizes` og `picture` lader browseren vælge mellem kandidatfiler.
- RSS 2.0 Specification — `image`-elementet, de tre formater det accepterer, og dets størrelsesgrænser.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

**Se, hvad det koster** Kontakt os →

# Gør begreberne praktiske

Spørg, hvordan »Hvad er WebP?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

**Se vores ydelser**

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

**Lavet med kærlighed af D.**

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

**Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.**

**Udskrevet fra https://engineer.company/da/glossary/webp/**