Engineer

WebP

растровий формат зі стисненням із втратами й без них

Що це таке

WebP — це растровий формат зображень від Google, який зберігають у файлах із розширенням `.webp` і віддають із зареєстрованим медіатипом `image/webp`. Один формат виконує роботу, для якої раніше були потрібні три: фотографії, що йшли як JPEG; пласка графіка з чіткими краями й прозорістю, що йшла як PNG; і короткі анімовані цикли, що йшли як GIF. Назву пишуть із великими літерами W і P, а на диску це просто файл `.webp`.

Всередині файл WebP — це контейнер RIFF, тобто та сама блокова структура, що й у звукових файлів WAV. Він підтримує два бітові потоки зображень. Блок `VP8` несе дані зі втратами, закодовані методом ключових кадрів відеокодека VP8: кодер передбачає кожен блок пікселів за сусідами, які вже декодовано, і зберігає лише різницю. Блок `VP8L` несе дані без втрат від окремого кодера, створеного для зображень, що мають повернутися біт у біт. Заголовок `VP8X` позначає розширену форму, яка може додавати альфа-канал, кадри анімації, колірний профіль ICC і метадані EXIF чи XMP. Google задокументувала формат разом зі своєю бібліотекою, а згодом його опублікували як RFC 9649, інформаційний RFC від листопада 2024 року.

Саме розширена форма робить формат чимось більшим, ніж просто менший JPEG. Зображення зі втратами може мати окремий альфа-канал, тож фотографія з м’яким прозорим краєм уже не мусить бути важким файлом без втрат. Анімація лежить у тому самому контейнері, кадр за кадром, і кожен кадр кодують зі втратами чи без них — як краще пасує.

Бітові потоки зображень мають власні межі. Кожну сторону закодованого зображення чи кадру записано в 14 бітах, тож вона сягає щонайбільше 16 383 пікселів для даних зі втратами і 16 384 для даних без втрат. Розширена форма може розміщувати кадри на більшому полотні, а весь файл

лишається трохи меншим за 4 ГіБ. Кожен канал займає 8 бітів: цього досить для зображень, призначених для екрана, але додаткова глибина 16-бітного оригіналу при цьому втрачається.

Застосуйте поняття до рішення

Застосуйте «Що таке WebP?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Інтерфейси, що швидко завантажуються, працюють на телефоні й поводяться однаково в усіх браузерах ваших клієнтів.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала full-stack-розробкою GIS-карт — full-stack-картографія — від бази даних до браузера

Керувала full-stack-розробкою GIS-карт Зв'язатися з нами →

Чому це важливо

Зображення зазвичай є найважчими байтами на сторінці, і саме формат визначає, скільки їх буде. Власне порівняння Google показує, що WebP без втрат на 26% менший за PNG, а WebP зі втратами на 25–34% менший за JPEG за однакового SSIM — міри структурної подібності двох зображень. WebP зі втратами та альфа-каналом зазвичай займає третину від PNG, який він заміняє. Менше байтів швидше доходить до екрана на повільному з'єднанні, дешевше обходиться в доставленні й лишає в бюджеті сторінки місце для того, заради чого прийшов читач.

Рішення, яке постає перед командою, стосується не так того, чи вживати WebP, як того, який формат випускати для якого зображення і чи випускати більше одного. Тут опубліковані середні значення перестають допомагати. Це середні по тестовому корпусу, тоді як файли справжнього сайту — це логотип, кілька фотографій і декілька знімків екрана, і порядок форматів для кожного з них може вийти іншим. AVIF, новіший формат, — звичний суперник, і на папері він стискає сильніше. На конкретному сайті, за тих розмірів, які цей сайт показує, він може програти.

Друге рішення — де відбувається перетворення. Якщо конвертувати вручну один раз, лишається тека файлів, які поступово розходяться зі своїми джерелами. Перетворення під час збірки, з оригіналу, що зберігається в репозиторії, робить кожен опублікований розмір відтворюваним і дає змогу згодом дешево змінити налаштування якості чи й сам формат.

Третє — усе, що поза браузером. Адреса зображення мандрує: у стрічки, електронну пошту, попередні перегляди в месенджерах, офісні документи та графічні редактори, і кожне з цих місць саме вирішує, які формати читати. Сайт, що переводить усі зображення у WebP, дізнається, хто з одержувачів його приймає, від першого читача, який побачить зламане зображення, — якщо тільки не перевірить кожен шлях заздалегідь.

Як це виглядає на практиці

Сторінка подяк описує зображення цього сайту як зображення WebP і PNG, розмір яких задається під час збірки. Шаблони показують, що це означає. Фотографії в тексті, аватари й знак бренду проходять через один спільний partial, який масштабує оригінал до кількох ширин, кожну кодує у WebP і пише один елемент `img` із `srcset`, тож ширину обирає браузер, а формат сайт уже обрав.

Цей формат — результат вимірювання, а не налаштування за замовчуванням. Версія Hugo, якою збирають сайт, уміє кодувати AVIF, тож віддавати обидва формати через елемент `picture` було можливо. У порівнянні, задокументованому 17 серпня 2026 року, за розмірів, які тоді показував сайт, AVIF вийшов на 30–36% більшим за WebP для знака з мозком і на 3–7% більшим для аватарів; він виграв лише на портреті, на 13%, і ледь-ледь на фотографіях офісу, на 3–4%. Елемент `picture` обирає джерело за порядком і за тим, що підтримує браузер, а не за розміром, тож якби AVIF стояв першим, кожен сучасний браузер отримував би приблизно на третину більше байтів за логотип на кожній сторінці. Коли варто випускати лише один формат, шаблон використовує `img` із `srcset` і зовсім обходиться без `picture`.

Той самий partial містить ще один урок. Hugo не пропускає зображення без змін, коли його просять змінити розмір до ширини, яку воно вже має: він перекодовує його, створюючи ще один опублікований файл, а для оригіналу зі втратами — ще один прохід зі втратами. Тому, коли запитана ширина дорівнює ширині оригіналу, partial за замовчуванням посилається безпосередньо на оригінал. Якщо виклик явно вимагає похідного файлу для кожного розміру, зображення все ж перекодовують; шорткод photo обирає цей шлях, щоб не публікувати другу копію оригіналу, який уже доступний через монтування assets.

WebP закінчується там, де закінчується те, що читають інші програми. Стрічка RSS вказує у своєму елементі `image` на PNG, бо RSS 2.0 приймає там лише GIF, JPEG і PNG та обмежує зображення розміром 144 на 400 пікселів: картка для поширення у WebP, завширшки 1280 пікселів, отримала попередження від валідатора стрічок W3C, і її місце зайняла піктограма apple-touch на 144 пікселі — PNG рівно максимальної ширини. Натомість знімки екрана, які англійський і данський маніфести передають до діалогу встановлення в Chromium, — це WebP, широкий і вузький для кожного, а задача lint падає, коли записана мітка їх генерації передує закомічений зміні у відстежуваних вхідних даних відтворення; незакомічені зміни ця перевірка не охоплює.

Де виникають проблеми

Перетворений JPEG двічі втрачає деталі. Коли наявний JPEG переводять у WebP зі втратами, декодується одне покоління зі втратами й кодується наступне, і артефакти обох лишаються. Перекодування в Hugo за тієї самої ширини — тихіша форма тієї самої пастки. Зберігайте в репозиторії оригінал без втрат або в повній якості й кодуйте кожен опублікований файл саме з нього.

Кольори тьмяніють після перетворення. `cwebp`, еталонний кодер, копіює метадані лише тоді, коли його про це попросити, тож профіль ICC фотографії з широким колірним охопленням губиться, і браузер трактує пікселі як sRGB. Додайте `-metadata icc` або перетворіть зображення на sRGB перед кодуванням.

Програма для читання стрічок показує зламане зображення. Програми, написані до появи формату або прив’язані до старішої специфікації, сприймають WebP як невідомий файл. RSS 2.0 — найочевидніший випадок: у ній елемент `image` називає три формати, і WebP серед них відсутній. Простежте кожне місце, куди мандрує адреса зображення, і віддавайте PNG чи JPEG там, де цього вимагає одержувач.

Довгий знімок екрана зупиняє кодер. Знімок цілої сторінки довгої статті легко перевищує 16 383 пікселі заввишки. Кодер libwebp обмежує кожну сторону цим розміром, хоча бітовий потік без втрат може вмістити на один піксель більше. Розділіть знімок на частини або спершу зменште його.

Більший файл перемагає в елементі `picture` **.** Новіший формат, поставлений першим, надходить до кожного браузера, що його декодує, незалежно від того, чи він менший для цього зображення. Перш ніж додавати джерело, виміряйте кожного кандидата за тих розмірів, які показує сторінка, як цей сайт зробив з AVIF.

Що почитати далі

- WebP — сторінка Google про формат: дослідження стиснення, з якого взято опубліковані відсотки, бібліотека libwebp та її інструменти.
- RFC 9649: WebP Image Format — контейнер, бітові потоки зі втратами й без них, розширені можливості та обмеження розмірів у вигляді інформаційного RFC.
- cwebp — усі параметри еталонного кодера, зокрема `-q` , `-lossless` і `-metadata` .
- Hugo: Image processing — як сайт на Hugo масштабує й перетворює зображення, зокрема у WebP, під час збірки.
- HTML Standard: Images — як `srcset` , `sizes` і `picture` дають браузерові обрати з-поміж файлів-кандидатів.
- RSS 2.0 Specification — елемент `image` , три формати, які він приймає, і його обмеження розміру.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке WebP?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/webp/