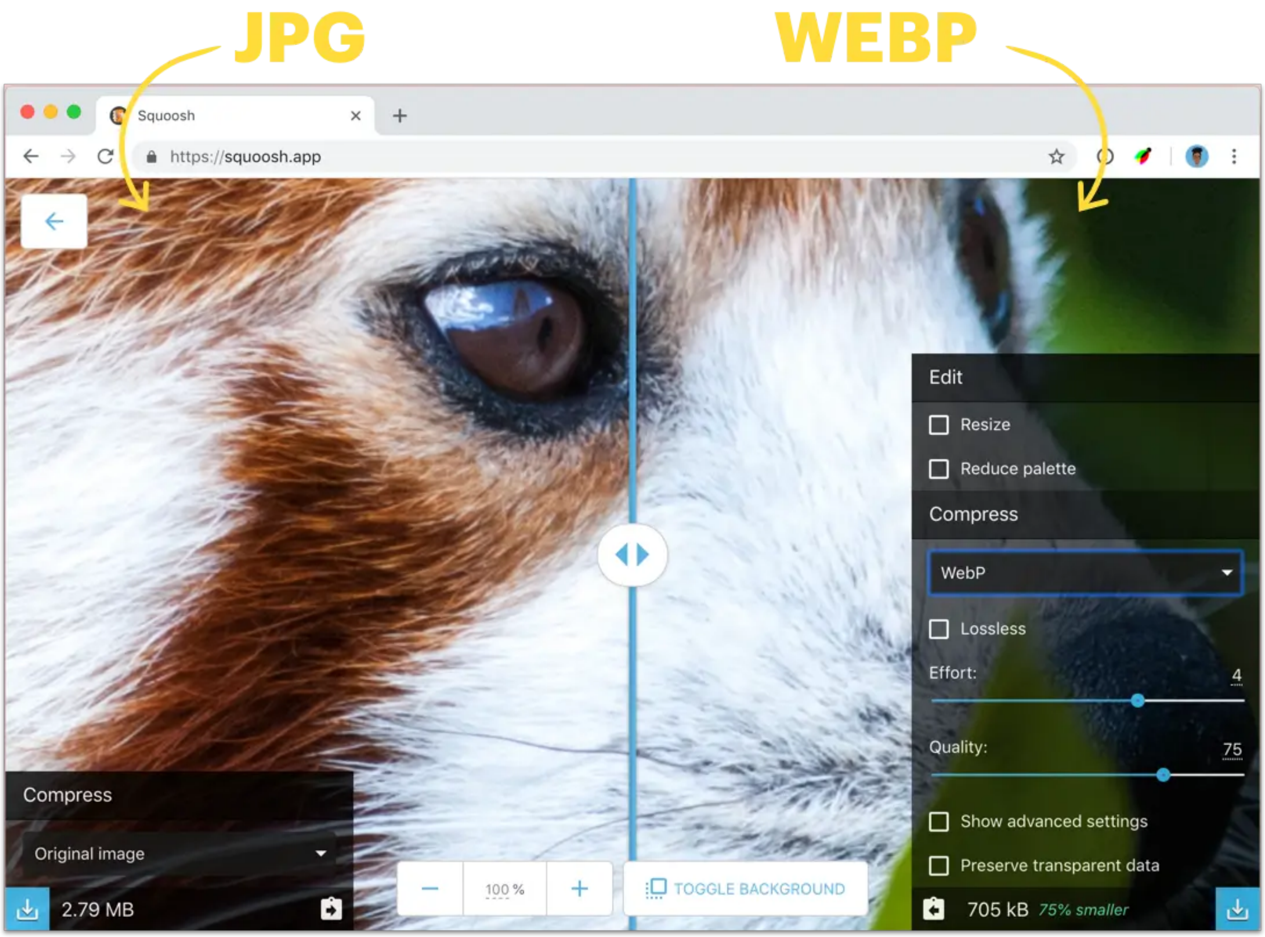
bitsofcode

Why and how to use WebP images today

Nov 21, 2018 *html, performance*

WebP is an image format developed by Google in 2010. It was created to be an alternative to formats like PNG and JPG, producing much smaller file sizes while maintaining similar image qualities.



Compress: Original image

2.79 MB

Compress: WebP

□ Lossless

Effort: 4

Quality: 75

□ Show advanced settings

□ Preserve transparent data

705 kB *75% smaller*

Why use WebP?

WebP is an incredibly useful format because it offers both performance and features. Unlike other formats, WebP supports both lossy and lossless compression, as well as transparency and animation.

	WebP	PNG	JPG	GIF
Lossy compression	✓	✓	✓	✗
Lossless compression	✓	✓	✓	✓
Transparency	✓	✓	✗	✓
Animation	✓	✗	✗	✓

Even with these features, WebP provides consistently smaller file sizes than its alternatives. In a comparative study of these image formats, it was found that WebP lossy images were on average 30% smaller than JPGs and WebP lossless images were on average 25% smaller than PNG.

How to covert images to the WebP format

There are several tools we can use to convert our JPGs, PNGs, and other file formats to WebP.

Online tools

- **Squoosh** - Online image compression and conversion
- **Online-Convert.com** - Online image conversion

Command line tools

cwebp is the most popular command line tool for converting images to the WebP format. Once installed, we can use it to convert images by passing in, among other options, a quality, input file, and output file.

```
# cwebp -q [quality] [input_file] -o [output_file]

cwebp -q 75 image.png -o image.webp
```

Node tools

imagemin, and it's plugin **imagemin-webp**, is the most popular node library for converting images to the WebP format.

Below is an example script that will convert all PNG and JPG files in the directory to WebP.

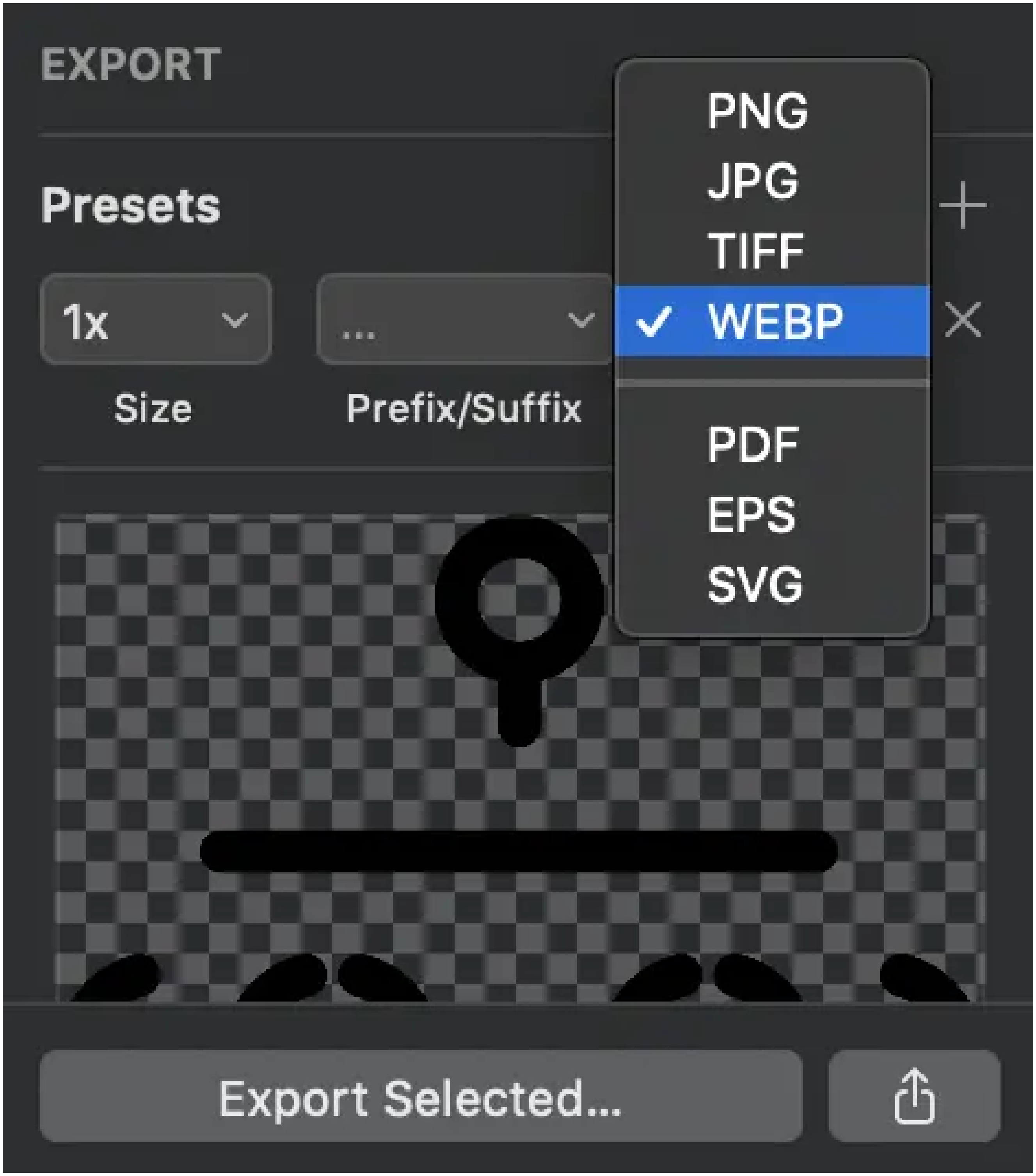
```
/* convert-to-webp.js */

const imagemin = require("imagemin");
const webp = require("imagemin-webp");

imagemin(["*.png", "*.jpg"], "images", {
  use: [
    webp({ quality: 75})
  ]
});
```

We can then use this script via the command line, or via build tool

```
node convert-to-webp.js
```

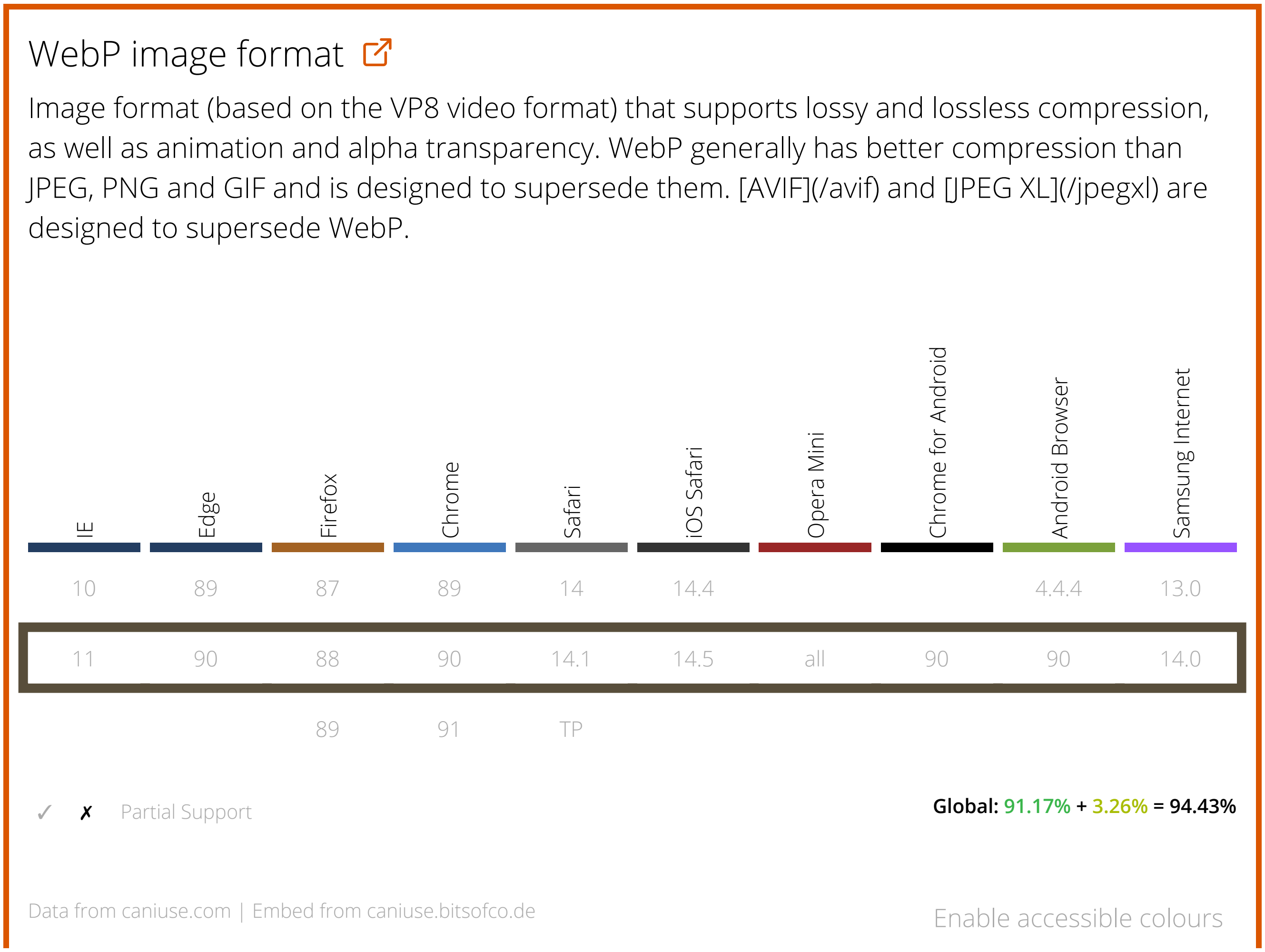
Sketch

In Sketch, we can export any slice in the WebP image format.



How to use to WebP on the web today

At the time of writing, WebP is supported in 72% of browsers globally used.



WebP image format

Image format (based on the VP8 video format) that supports lossy and lossless compression, as well as animation and alpha transparency. WebP generally has better compression than JPEG, PNG and GIF and is designed to supersede them. [AVIF](/avif) and [JPEG XL](/jpegxl) are designed to supersede WebP.

IE	Edge	Firefox	Chrome	Safari	iOS Safari	Opera Mini	Chrome for Android	Android Browser	Samsung Internet
10	89	87	89	14	14.4			4.4.4	13.0
11	90	88	90	14.1	14.5	all	90	90	14.0
		89	91	TP					

✓ ✗ Partial Support

Global: 91.17% + 3.26% = 94.43%

Data from caniuse.com | Embed from caniuse.bitsofco.de

Enable accessible colours

Although this is good enough to make a compelling case to use WebP, it isn't good enough to rely on the format without providing fallbacks. In browsers that don't support this file format, the image will appear broken.

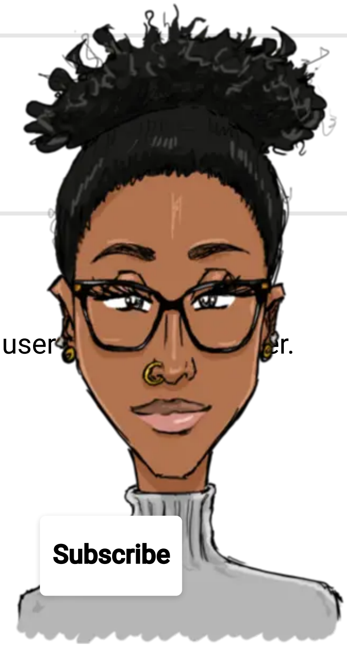
We can provide fallbacks to images using the **`<picture>`** element. This HTML5 element allows us to provide multiple sources for a single image.

```
<picture>
    <source type="image/webp" srcset="image.webp">
    <source type="image/jpeg" srcset="image.jpg">
    <img src="image.jpg" alt="My Image">
</picture>
```

To provide an alternate image source, we use the **`<source>`** element within the **`<picture>`** element. The **`<source>`** element has a number of attributes we can use to define the image and when it should be used:

- `type`: The MIME type of the source
- `srcset`: The path to the image file. Multiple files can be used to provide different image sizes/densities (see Responsive Images - The srcset and sizes Attributes)
- `sizes`: A list of sizes of each source file (see article above)
- `media`: A media query that will determine when the image is used (see article above)

In addition to the various `<source>`s, a regular `<img>` element must also be provided as a fallback for browsers that do not support multiple file formats via the `<picture>` element.



Subscribe to the Newsletter

Receive quality articles written by Ire Aderinokun, frontend developer and user r.

Email Address*

First Name

Subscribe

←

Web workers vs Service workers vs Worklets

javascript

→

Using a headless browser to capture page screenshots

javascript

Comments Community Privacy Policy 1 Login

♡ Recommend 6 Tweet Share Sort by Best

Join the discussion…