

A Measured Image Budget for the Web

Choose slots, count actual bytes and keep visual decisions honest

01

02

03

How to use this guide

People preparing image assets for a small web page.

An original three-asset workshop page combines actual SVG byte measurements, a responsive-width planning table and an explicit visual-review handoff.

- Inventory images by reader purpose.
- Measure real saved SVG bytes without inventing raster exports.
- Calculate responsive candidate coverage and identify a missing width.
- Separate local structure checks from browser, visual and network acceptance.

Scope and limitations

- The measured fixtures are original constrained SVGs, not photographic encoding benchmarks.
- The raster ledger contains unperformed export plans, not measured compression results.
- No speed score, browser selection, visual equivalence or device result is guaranteed.

Published by KitForma. Edition 1.0. Sources checked 1 October 2026. People and businesses in worked examples are fictional. This is an independent publication, not endorsed by the platforms mentioned.

Copyright 2026 KitForma. Usage, support and refund terms are included in LICENSE.txt in the purchased package.

Contents

How to use this guide	2
Contents	3
Start with a page and a reason for each image	4
Sources and tools	6

01

Start with a page and a reason for each image

The fictional Field Notes Club is publishing a small page about a paper-folding workshop. It needs a folding-sequence diagram, a comparison of two paper sizes, and a simple club mark. The page has no photography, advertisements, analytics screenshots, or product reviews. This deliberately small case makes every asset explainable. Your task is to prepare a measured image package, identify what the measurements establish, and write down the remaining display checks before the page is released.

Start by listing the reader's questions. Which fold comes first? Which sheet size is needed? Is this the correct workshop page? The diagram, size comparison, and mark answer those questions in that order. A decorative texture may look pleasant, but it is not required to answer them. The fastest asset is sometimes one you decide not to include. That decision should follow the page's purpose rather than an arbitrary rule that every page needs a large hero photograph.

The supplied sources are three original SVG files, meaning text descriptions of shapes and labels rather than photographs. Their small size and visible structure make them suitable for a reproducible local byte experiment. The checker produces compact SVG candidates, measures their actual saved byte lengths, and compares their shape-and-text structure. It does not export JPEG, WebP, or AVIF files. The accompanying raster ledger intentionally leaves unperformed export measurements empty instead of inventing attractive compression percentages.

A budget belongs to a defined delivery scenario. This workbook uses a fictional ceiling of 3,000 uncompressed image-file bytes for the three compact SVG assets combined. That is an exercise constraint, not a universal recommendation for production websites. It excludes HTML, fonts, headers, protocol overhead, and other resources. A separate cold-load network measurement would answer a different question. Keep the exercise ceiling, the local file measurement, and the eventual browser result in separate columns.

Sources and tools

[S1] MDN: Using responsive images in HTML

https://developer.mozilla.org/en-US/docs/Web/HTML/Guides/Responsive_images

Checked: 2026-10-01

[S2] web.dev: Automating compression and encoding

<https://web.dev/learn/images/automating/>

Checked: 2026-10-01

[S3] W3C: An alt decision tree

<https://www.w3.org/WAI/tutorials/images/decision-tree/>

Checked: 2026-10-01

Inside the complete guide

Inventory the sources before making candidates

Translate display slots into candidate requirements

Choose a representation that preserves the message

Measure bytes and preserve a meaningful baseline

Treat readability and cropping as acceptance decisions

Give the layout honest dimensions and loading priorities

Write alternatives that work when the image does not

Build a release ledger someone else can reproduce

Capstone: release a measured package without invented results