TAPFLARE / RESEARCH & PRACTICAL GUIDANCE

Landing Page Image Requirements: Formats, Crops and Loading

> Tapflare is a flat-rate monthly subscription for graphic design and website development. A dedicated project manager assigns each request to senior designers and developers, so companies get steady creative and web output without hiring.

Published by Tapflare · 2026-11-01 · landing page image requirements · image handoff checklist · responsive images · hero image dimensions · image optimization · webp vs avif · web design · image accessibility · website development

Original article: https://tapflare.com/articles/landing-page-image-handoff



In this article

1. Executive Summary
2. Introduction and Background
3. Define the Asset Inventory and Acceptance Rules
4. Specify Slots, Crops and Format Policy
5. Assign Responsive Derivatives to the Platform
6. Set Loading, Alternatives and Review Ownership
7. Build the Image Manifest and Verify the Page
8. Data Analysis and Evidence
9. Case Studies and Real-World Examples
10. Implications and Future Directions
11. Conclusion

Executive Summary

Landing page image requirements are a shared delivery specification: what each image communicates, where it appears, which crop preserves its meaning, how responsive files are generated, and who approves the published result. Start with the **display slot**, the space the image occupies in the layout, rather than a universal hero-image dimension. Responsive sizing describes that slot; art direction changes the composition for a different context. (developer.mozilla.org)

The delivery pack should contain an editable or high-quality source master, approved crop instructions, a manifest of intended uses, content-alternative decisions, and implementation evidence. Choose photographic, interface and vector assets separately. **AV1 Image File Format (AVIF) and WebP** both support lossy and lossless compression, so a filename extension alone does not establish acceptable visual quality. (developer.mozilla.org) Assign early discovery to the likely **Largest Contentful Paint (LCP)** image and defer suitable offscreen assets. Google explicitly advises against lazy-loading the LCP image. (web.dev)

Platform automation must be specified by workflow. Webflow documents responsive-image exceptions, including rich-text insertion; Shopify recommends checking the actual downloaded source across viewport sizes. Those are reasons to name the upload route and verification owner in the brief. (help.webflow.com) (shopify.dev) The proposed manifest in this report connects source files, display slots, focal points, derivative ownership, formats, candidate widths, alternatives, loading and acceptance evidence. It is a commissioning template, not an established technical standard.

For scale, an illustrative slot **600 Cascading Style Sheets (CSS) pixels wide at device-pixel ratio 2** suggests evaluating a **1,200-pixel candidate** under the documented density relationship; this is arithmetic, not a universal export prescription. (developer.mozilla.org) For published-page assessment, the good LCP threshold is **2.5 seconds or less at the 75th percentile**, assessed separately for mobile and desktop. (web.dev) The good **Cumulative Layout Shift (CLS)** target is **0.1 or less**. (web.dev) These are page-experience measures, not individual-image byte budgets. The final sign-off should pair visual review with implementation evidence and an alternative that conveys the image’s meaning in context. (www.w3.org)

Introduction and Background

A landing page image can be visually approved and still arrive with an incomplete implementation brief. A designer may deliver a wide photograph without identifying the mobile crop. A developer may generate smaller files while preserving a composition that no longer communicates the intended subject. A marketing reviewer may approve the source file without seeing the version visitors receive.

The practical question is therefore broader than “What is the best image format for landing pages?” It is whether the handoff describes the image’s job, the layout it must serve and the evidence needed to accept delivery. This report treats **landing page image optimization as a coordination task** supported by browser and platform documentation.

General design guidance already covers compression, modern formats and correct dimensions. ManyPixels includes those tasks in its broader web-design rules. (www.manypixels.co) The additional proposal here is an asset-by-asset agreement that connects those tasks to named owners and acceptance conditions. That proposal is a workflow hypothesis.

The scope is static landing-page photography, screenshots, illustrations, logos and decorative imagery. It excludes a general graphics-programming tutorial. The browser also retains discretion over responsive-source selection, so the specification should define suitable choices and verify behavior rather than promise that every visitor downloads an identical file. (html.spec.whatwg.org)

Scope note: No client page or sample exports were supplied, and no file exports or page-performance measurements were made for this report. It provides **documented mechanisms, illustrative calculations and a fictional handoff**; it makes no claims of measured savings, conversion results, delivery-time improvements or performance gains. In a real project, request timing can be inspected in the browser’s Network waterfall; that is evidence about the tested delivery, not an assumption drawn from the design file. (developer.chrome.com) The goal is a reviewable agreement that marketing, design and implementation teams can use before launch.

Define the Asset Inventory and Acceptance Rules

Inventory images by **purpose**, not just filename or folder. An informative image’s alternative should convey its meaning in the surrounding content; that decision depends on why the image is present. (www.w3.org) Apply the same purpose-first approach to visual approval.

A useful starting inventory distinguishes:

- **Hero photograph:** Record the subject, visual emphasis and space reserved for the headline. Identify what must remain visible when the layout narrows.
- **Product screenshot:** Record which interface detail supports the claim. Name the labels or controls that must remain readable in the approved presentation.
- **Decorative background:** Record the intended visual treatment and explicitly assign the decision about whether it adds information.

- **Logo or icon:** Request an appropriate vector master when the artwork is geometric and must scale cleanly. **Scalable Vector Graphics (SVG)** represents drawing instructions rather than a fixed photographic pixel grid. (developer.mozilla.org)
- **Below-fold illustration:** Record the explanatory purpose, associated copy and reviewer, rather than assuming placement alone determines its meaning.

Next distinguish **source master**, **approved crop** and **delivery derivative**. These are proposed workflow labels. The master preserves editing options; the crop records a composition; the derivative is the file delivered for a particular rendering condition. A single master may support several derivatives without supporting every possible composition.

For raster artwork, preserve the most useful available source before final delivery compression. The **Joint Photographic Experts Group (JPEG)** format lacks an alpha channel; **Portable Network Graphics (PNG)** can support lossless graphics and transparency. That makes the content and background requirements relevant to the export decision. (developer.mozilla.org) Do not choose a photographic format simply because the screenshot and photograph share a folder.

The inventory should also capture **provenance and usage conditions**. A stock-image license does not automatically establish every permission for every proposed use. Unsplash's guidance, for example, distinguishes the image license from other permissions that may be relevant. (help.unsplash.com) Keep the source URL, license record, attribution requirements and confirmation of the intended use with the master.

The acceptance rule should describe the result in ordinary language: "the product label remains readable," "the subject remains inside the visible crop," or "the background introduces no information absent from the copy." These proposed criteria help reviewers judge the delivered asset against its job.

Specify Slots, Crops and Format Policy

Start with the layout and focal subject

A **display slot** is the image's rendered space in the layout, measured in CSS pixels. The **srcset** attribute lists candidate sources; with width descriptors, the **sizes** attribute communicates the expected slot under relevant media conditions. It should describe the layout rather than simply repeat the uploaded image's width. (developer.mozilla.org)

Ask the designer and developer to agree on:

- **Slot behavior:** Full-width, constrained column, split hero, card or background.
- **Aspect ratio:** Whether the visible box keeps its proportions or changes across layouts.
- **Focal subject:** The feature that must remain visible and the direction in which cropping is acceptable.
- **Exclusion area:** Space intended for live text, interface controls or overlapping content.
- **Crop approval:** Whether one composition works throughout or separate approved compositions are required.
- **Quality criterion:** Which details must remain legible or visually clean at the intended display size.

The distinction between **resolution switching** and **art direction** matters. Resolution switching supplies different pixel quantities for essentially the same composition. Art direction supplies a changed composition, such as a closer mobile crop; the picture element supports this use. (developer.mozilla.org) CSS object-fit with cover can fill a box while clipping content when the aspect ratios differ. Filling the box therefore does not prove that the subject survived. (developer.mozilla.org)

For candidate selection, the **HyperText Markup Language (HTML) Standard** relates effective density to candidate width divided by source size. This supports sizing from the intended slot and density instead of selecting a universal hero width. (html.spec.whatwg.org) Record candidate dimensions as a delivery policy to review, not a substitute for checking the rendered result.

Choose a content-specific format policy

WebP versus AVIF is a comparison of actual outputs and delivery conditions. Both support lossy and lossless encoding, so the required visual fidelity remains a separate decision. (developer.mozilla.org) Imgix explicitly describes AVIF size savings as a statistical average across requests, rather than a guarantee for every asset. (docs.imgix.com)

Use these practical commissioning rules:

- **Photography:** Compare suitable encodings using the approved crop and intended viewing size.
- **Interface screenshots:** Examine small text, thin rules, gradients and edges; reject an export that compromises the detail the image must explain.
- **Vector artwork:** Preserve the editable vector source and identify whether the delivery path retains it or requires a raster derivative.
- **Transparency:** Test the chosen asset over its actual page background, including edge appearance.
- **Fallback policy:** Name the delivery mechanism and tested browser requirements instead of assuming the newest codec is always the chosen response. Imgix documents an automatic format sequence based on browser support. (docs.imgix.com)

A focal point is also implementation data. Imgix’s crop documentation describes a centered default and controls for alternate alignment. (docs.imgix.com) If the handoff merely says “crop automatically,” it leaves a consequential visual decision unspecified.

Assign Responsive Derivatives to the Platform

The designer should supply sufficient source quality and approved compositions. The implementation owner should state what the **content management system (CMS)**, theme or **content delivery network (CDN)** generates. Upload acceptance, resizing, format negotiation and art direction are separate capabilities.

Table 1 summarizes documented behavior relevant to an image handoff, as checked for this report on **November 1, 2026**. This is a retrieval date, not a claim that every installed site uses the latest documented behavior. The final column identifies the project-specific decision still required.

Platform	Documented derivative behavior	Handoff decision and verification
Webflow	Responsive variants are documented for supported inline image workflows, with rich-text exceptions. Upload acceptance and compression are documented separately. (help.webflow.com)	Name the upload route and supported source type. Require the developer to inspect generated variants rather than assume every accepted format follows the same path.
WordPress	Generated image markup supports responsive candidates; additional registered sizes depend on the theme. Its classic-theme documentation describes same-format subsizes by default. (developer.wordpress.org)	Name the theme, registered crops and active conversion path. Verify the deployed installation rather than assuming a universal format-conversion policy.
Shopify	The image_tag filter can generate default candidate widths; image_url supports crop requests and cannot enlarge beyond source dimensions. (shopify.dev)	Name the theme's slot and crop rules. Supply sufficient source resolution and verify the rendered request.
Framer	Optimization documentation describes responsive attributes for most image elements and WebP delivery where AVIF is unsupported. (www.framer.com)	Record the image resolution setting and inspect the published result, including the actual chosen crop and file.

The common pattern is **conditional automation**. The table describes documented mechanisms, not a guarantee about a particular page. Separate creative approval from the implementation path used to produce it.

Webflow specifically says comma-separated value (CSV) imports and application programming interface (API) uploads do not create responsive variants. (help.webflow.com) Its breakpoint aspect-ratio controls can also change the box around the same image; they do not establish that a designer supplied a separately composed mobile master. (help.webflow.com)

For WordPress, a developer can register proportional sizes or specified crop positions. (developer.wordpress.org) Ask for that policy in the handoff rather than requesting an unexplained collection of exports. A crop that works in one theme component may be unsuitable for another.

Shopify documents a focal point shared wherever the image is used, with theme support requiring attention. (help.shopify.com) Framer distinguishes responsive Auto settings from fixed image-size settings. (www.framer.com) Framer also states that it never upscales images, reinforcing the need for sufficient source resolution. (www.framer.com) Both examples show why the manifest should name the component or setting, not just the platform.

For a separate CDN workflow, Cloudflare documents resizing, cropping and transcoding from source images, while Imgix documents browser-dependent automatic format selection. (developers.cloudflare.com) (docs.imgix.com) Assign the transformation policy to the developer and the visual acceptance to the designer. Do not ask both parties to generate competing derivative sets without specifying which set the page will use.

Set Loading, Alternatives and Review Ownership

Separate early content from deferrable imagery

Largest Contentful Paint concerns the largest eligible visible content element, which can be an image, text block or video. The hero photograph is therefore a candidate to investigate, not an automatic identity for the measured element. (web.dev)

For a likely LCP image, Google recommends early discovery in the initial HTML document. (web.dev) Its guidance also recommends **fetchpriority="high"** for a likely LCP image and says not to lazy-load it. (web.dev) The implementation brief should name the candidate and require a trace to confirm which element actually matters at representative viewports.

Use a loading policy with distinct decisions:

- **Likely main-content image:** Make it discoverable early and test its actual request timing.
- **Offscreen content:** Consider native lazy loading when the image is not needed immediately.
- **Responsive exception:** Reassess when a layout change moves an asset into the initial viewport.
- **Evidence owner:** Require the developer to identify the loaded source and capture the relevant request behavior. Chrome can expose both initial and final fetch priority. (developer.chrome.com)

Native lazy loading defers requests until images approach the viewport, rather than waiting for a universal scroll position. (web.dev) It is therefore unsuitable as a blanket instruction for all landing-page images.

Give alternatives and approval explicit owners

An image alternative is a content decision. W3C guidance ties it to the image’s purpose: informative imagery communicates meaning, functional imagery communicates an action or destination, and purely decorative imagery uses an empty alternative. (www.w3.org)

Table 2 proposes a role split. These are project responsibilities to agree, rather than a claim that a platform supplies the approvals automatically.

Role	Decision to own	Acceptance evidence
Marketing/content owner	What the image means, which claims it supports, and whether surrounding copy conveys the necessary information.	Approved alternative or decorative decision, tied to the page copy.
Visual designer	Composition, focal subject, crop, source quality and acceptable appearance.	Approved previews at the specified layout states.
Developer	Markup, responsive candidates, reserved space, loading and transformations.	Published-page source and request evidence.
Project manager	Completeness, file versions, dependencies and recorded approvals.	Manifest showing a named owner and review status for each asset.
Asset rights owner	Provenance, usage terms and any required credits or permissions.	Source and license records linked to the master.

The division prevents an implementation decision from silently replacing a content or visual decision. For complex informative images, W3C recommends making the contained information available elsewhere on the page. (www.w3.org) Apply that principle when a dense product screenshot cannot communicate its essential explanation through a short alternative alone. When meaningful text appears only inside an image, the decision tree calls for including that text in its alternative. (www.w3.org)

License details should travel with the pack. **Creative Commons Attribution 4.0** requires appropriate credit, a license link and an indication of changes. (creativecommons.org) The standard Unsplash license permits covered commercial uses without permission and does not require attribution, although it appreciates it. (unsplash.com) Those are different conditions; do not infer one asset's obligations from another asset's folder.

Build the Image Manifest and Verify the Page

The proposed **image manifest** connects visual delivery and implementation. Copy it into a tracker or spreadsheet, and mark unresolved decisions pending.

Table 3 is a **Hypothetical Example**. Slots use illustrative CSS pixels; width candidates use source pixels. Review these candidates against the responsive-source model. (html.spec.whatwg.org)

ID / purpose	Source master	Display slots	Crop / focal point	Derivative owner	Format / fallback policy	Width candidates	Alternative owner / decision	Loading	Test evidence
HERO / photograph	Original photo plus approved crops	Desktop 600 × 400; mobile 360 × 450	Product stays visible; separate mobile crop	Developer through selected delivery path	Compare AVIF/WebP; retain supported fallback	Desktop 600, 1200; mobile 360, 720	Content owner: concise purpose-specific alternative	Early; check actual LCP	Pending crop approval and source/request capture
UI / screenshot	Editable interface capture source	Desktop 600 wide; mobile detail 360 wide	Preserve the relevant controls; separate detail view	Developer; designer approves output	Evaluate lossless output; record chosen delivery type	600, 1200; detail 360, 720	Content owner: explanation beside image plus suitable alternative	Depends on actual placement	Pending text-legibility review
BG / decoration	Editable illustration	Section-sized background	Keep safe area around live heading	Developer	Approved raster or vector path	Set after section layout	Content owner: decorative decision	Defer only if offscreen	Pending overlap and loading check
LOGO / identity	Approved vector master	Fixed header slot; responsive layout	Keep full mark and required spacing	Designer supplies vector; developer integrates	SVG where supported by chosen route	Vector; raster fallback only if needed	Content owner: context-dependent alternative	Initial header content	Pending visible and accessible-purpose review

ID / purpose	Source master	Display slots	Crop / focal point	Derivative owner	Format / fallback policy	Width candidates	Alternative owner / decision	Loading	Test evidence
ILL / explanation	Editable illustration	Content-column width	Retain explanatory detail	Developer with designer approval	Compare appropriate vector/raster delivery	Set after slot confirmation	Content owner: meaning and supporting copy	Lazy if suitably offscreen	Pending crop, source and request evidence

The table exposes missing decisions. “Pending” records acceptance status. A CDN can generate derivatives, but transformation capabilities do not supply creative approval. (developers.cloudflare.com)

For each representative layout, use this **landing page image handoff checklist**:

- **Match the slot:** Compare the rendered image box with the agreed layout dimensions.
- **Inspect the crop:** Check the focal subject, safe area and all meaningful edges.
- **Identify the source:** Record the selected responsive image URL and its pixel dimensions.
- **Check transfer:** Record the actual response and transferred bytes, with cache conditions noted.
- **Check timing:** Inspect when the request starts and its priority relative to other work.
- **Check stability:** Confirm that appropriate space is reserved before the image finishes loading.
- **Check meaning:** Review the alternative, nearby copy and any linked action.
- **Record sign-off:** Attach the evidence and name the reviewer for each unresolved issue.

Shopify recommends checking downloaded sizes across viewports. (shopify.dev) Chrome’s Network tools support request-timing inspection; image width and height attributes help reserve layout space. (developer.chrome.com) (developer.mozilla.org) Preserve visual approval alongside these technical observations.

Data Analysis and Evidence

Dimensions are a calculation, not a universal recommendation

Consider a **Hypothetical Example** with a slot **600 CSS pixels wide** and target **device-pixel ratio (DPR) 2**. DPR expresses the relationship relevant to source density. Multiplying the illustrative slot by density gives **600 × 2 = 1,200 source pixels**. The example follows MDN’s explanation of device pixels per CSS pixel, rather than establishing a required export dimension. (developer.mozilla.org)

For the same hypothetical composition at a slot **360 CSS pixels wide**, the corresponding arithmetic is **360 × 2 = 720 source pixels**. The HTML Standard’s effective-density relationship explains the calculation; it does not compel every browser to request the same candidate. (html.spec.whatwg.org) The manifest should retain enough source information to revise candidates after inspecting actual output.

Transformation services also have their own parameter rules. Imgix requires a width, height or both when using its DPR parameter. (docs.imgix.com) Cloudflare describes DPR as a multiplier of output resolution. (developers.cloudflare.com) A project must therefore record whether density is expressed through candidate

widths, a service parameter or another documented delivery path.

Page thresholds and evidence limits

The documented **good LCP threshold is 2.5 seconds or less**, evaluated at the **75th percentile** of page loads and separately across mobile and desktop. (web.dev) The good **CLS threshold is 0.1 or less**, with field assessment concerned with the distribution of page visits. (web.dev) These provide page-level context; they do not imply a maximum acceptable byte count for each asset.

A default Lighthouse load audit may miss shifts that happen later in a page's life. (web.dev) Chrome's Performance panel distinguishes local measurements from available Chrome UX Report field experience. (developer.chrome.com) Keep the categories explicit:

- **Lab evidence:** A reproducible test under recorded conditions, useful for investigating the delivered page.
- **Field evidence:** Measurements from real visits, with the population and reporting scope identified.
- **Visual evidence:** Human review of the selected delivery at its actual display size.
- **Export evidence:** The file, encoder settings, dimensions and byte count used in a comparison.
- **Unknowns:** Missing coverage or untested conditions recorded without converting them into success claims.

A future comparison should state the file, source content, encoder settings, viewport, network/device environment, tool version, date and lab/field classification. AVIF's lack of progressive rendering is another documented format characteristic to consider; format choice still requires reviewing the actual delivery. (developer.mozilla.org)

If measuring a candidate, calculate byte saving as **(original bytes minus candidate bytes) divided by original bytes, multiplied by 100**, with the original clearly identified. A smaller file does not by itself establish a field LCP improvement or a business outcome.

Case Studies and Real-World Examples

A split hero with a missing mobile composition (Hypothetical Example)

A fictional software landing page places a product photograph beside live introductory copy. The supplied wide composition looks appropriate in the desktop design. The mobile component uses a taller box and fills it with cover behavior. Because cover can clip the image when box and image proportions differ, that implementation can remove part of the product. (developer.mozilla.org)

The initial instruction, "use the hero responsively," leaves the critical decision unresolved. A smaller version of the same wide image would change its pixel count without necessarily changing the visual composition. A picture-based art-direction approach can instead select a separately approved crop under relevant conditions. (developer.mozilla.org)

The resolved specification uses the hypothetical HERO row in Table 3. It preserves the original photograph, adds the approved mobile composition, names the product as the focal subject and leaves the heading as live page content. The developer owns candidate generation and source selection; the designer approves the visible crop.

If using a crop service, the developer records the chosen alignment instead of relying on an undocumented default. Imgix, for example, documents a centered default for its crop mode. (docs.imgix.com) If using Webflow, the developer also confirms that the chosen insertion workflow generates responsive variants. (help.webflow.com) For Framer, its instructional guidance calls for checking cropping at each breakpoint. (www.framer.com)

The content owner then decides whether the photograph adds information and approves its alternative accordingly. If an image instead supplies a linked action, the alternative should communicate the destination or action. (www.w3.org) The developer records the intended aspect ratio and verifies reserved space rather than accepting a visually correct final frame that shifts during loading. (web.dev)

The resolution is a complete, reviewable specification. Acceptance requires the approved composition, correct published source, appropriate loading and recorded sign-off. Browser-level inspection of downloaded sizes across layouts remains part of that acceptance, as Shopify's responsive-image guidance recommends. (shopify.dev)

Implications and Future Directions

The useful resourcing question is **who has the authority and time to finish each decision**. For teams commissioning through Tapflare or another design-and-development provider, the manifest can become a shared acceptance document. It should accompany the request and return with the delivered work, rather than remain an informal conversation.

The specification also needs a maintenance owner. A refreshed photograph, revised screenshot or different page layout should trigger review of its associated crop, derivative policy, meaning and evidence. Treat approval as attached to a particular source and context.

Automatic format delivery can reduce manual export work, but the brief should state the mechanism. Cloudflare documents fallback behavior during automatic or requested format conversion, while Framer documents WebP fallback where AVIF is unsupported. (developers.cloudflare.com) (www.framer.com) Neither statement justifies a promise that every delivery will use the same codec.

The most practical improvements are procedural:

- **Use stable asset IDs:** Link the source, manifest row and implementation component.
- **Approve compositions explicitly:** Keep unresolved mobile crops visible in the project tracker.
- **Retain test conditions:** Make later comparisons meaningful by preserving what was checked.
- **Recheck changed assets:** Carry the review forward when the source or layout changes.

Preserve provenance with the same care as technical settings. Creative Commons notes that its license may not supply all permissions for a proposed use, and Unsplash separately explains limitations around other rights. (creativecommons.org) (help.unsplash.com) Record the applicable decision rather than treating a downloaded file as complete permission evidence.

The manifest remains a proposed working method. Its usefulness should be evaluated through clearer assignments and fewer unresolved acceptance questions, without presenting those outcomes as established research findings.

Conclusion

A complete landing page image handoff connects **visual purpose, composition, delivery and verification**. It tells the designer what must survive a crop, tells the developer what the platform should generate and tells the content owner which meaning must remain available to visitors.

The decision sequence is practical: inventory the assets, define their display slots, approve the necessary compositions, preserve useful masters, assign derivative generation and specify loading. Add a named owner for alternatives and source permissions. Then review the published page using both visual and implementation evidence.

The proposed manifest turns those decisions into something a team can copy, assign and accept. An unresolved crop, missing alternative or unspecified upload route remains visible as a pending decision. That is more reviewable than a folder whose filenames imply completion.

The best format and dimensions follow from the asset’s content, intended layout and delivery mechanism. A photographic hero, detailed screenshot and vector logo should not inherit an identical export rule merely because they share a page. Likewise, loading policy should reflect when the image is needed.

For commissioning teams, the final deliverable is therefore more than an optimized file. It is a traceable set of source assets, approved compositions, implementation policies and acceptance evidence. The fictional example demonstrates how to complete that specification.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://developer.mozilla.org/en-US/docs/Web/HTML/Guides/Responsive_images
2. https://developer.mozilla.org/en-US/docs/Web/Media/Guides/Formats/Image_types
3. https://web.dev/articles/optimize-lcp
4. https://help.webflow.com/hc/en-us/articles/33961378697107-Responsive-images
5. https://shopify.dev/docs/storefronts/themes/best-practices/performance/use-responsive-images
6. https://web.dev/articles/lcp
7. https://web.dev/articles/optimize-cls
8. https://www.w3.org/WAI/tutorials/images/informative
9. https://www.manypixels.co/blog/web-design/rules-to-know

10. https://html.spec.whatwg.org/multipage/images.html
11. https://developer.chrome.com/docs/devtools/network/reference
12. https://help.unsplash.com/en/articles/2612329-releases-and-trademarks
13. https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/picture
14. https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Properties/object-fit
15. https://docs.imgix.com/en-US/apis/rendering/automatic
16. https://docs.imgix.com/en-US/apis/rendering/size/crop-mode
17. https://developer.wordpress.org/apis/responsive-images
18. https://shopify.dev/docs/api/liquid/filters/image_tag
19. https://www.framer.com/help/articles/how-are-images-optimized-in-framer
20. https://help.webflow.com/hc/en-us/articles/33961381659411-CSS-width-and-height
21. https://developer.wordpress.org/reference/functions/add_image_size
22. https://help.shopify.com/en/manual/online-store/images/theme-images
23. https://www.framer.com/help/articles/image-resolution-options
24. https://developers.cloudflare.com/images/optimization/transformations/overview
25. https://web.dev/learn/images/performance-issues
26. https://www.w3.org/WAI/tutorials/images/decision-tree
27. https://creativecommons.org/licenses/by/4.0/deed.en
28. https://unsplash.com/license
29. https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/img
30. https://docs.imgix.com/en-US/apis/rendering/device-pixel-ratio
31. https://developers.cloudflare.com/images/optimization/features
32. https://developer.chrome.com/docs/devtools/performance/overview
33. https://www.framer.com/academy/lessons/framer-fundamentals-adding-images-video-and-text

THE PUBLISHER

About Tapflare

Tapflare gives companies an on-demand design and web team for a flat monthly fee.

How it works

Each subscriber has a dedicated project manager who learns the brand, assigns every request to a senior specialist in that kind of work, and reviews the result before it is delivered. Requests are submitted and discussed in the Tapflare portal, each plan includes a daily allotment of design or development time, and the client keeps full ownership of the export and source files.

What it covers

Graphic design, custom illustration, presentation design, logos and brand guides, video editing and motion graphics, website and mobile-app mockups, and website development, from design mockups to HTML, to drag-and-drop builders or custom code.

What it is not

Tapflare is not a marketing agency, a media-buying, SEO or copywriting service, a software product or a freelance marketplace. Where a reader needs one of those, the honest answer is to say so.

Visit

See Tapflare and its pricing.

Articles published here explain general practice and cite their sources. Prices and plan contents, including Tapflare's own, change; check the provider's current pricing page before deciding.

Website: https://tapflare.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.