

PRODUCT PRE-ANNOUNCEMENT

# Coming Soon From Mandelbrot Metal: ChromaForge

---

*A developer-focused color system tool built from the palette technology behind Mandelbrot Metal.*

**Website:** [chromaforge.dev](https://chromaforge.dev)

**Supported devices:** iPhone, iPad, and Mac (iOS/iPadOS/macOS 26+)

**Audience:** Developers, web designers, app builders, and design-system teams

**Launch model:** Free to explore; Pro for production workflows

---

## Core promise

ChromaForge is being designed to help creators move from color inspiration to usable color infrastructure: palettes that can be previewed, repaired, tokenized, and exported into real development workflows.

Mandelbrot Metal began as a fast, beautiful fractal explorer, but one of its most powerful internal technologies has always been its palette engine.

Inside Mandelbrot Metal, color is not decoration. It shapes structure, depth, atmosphere, legibility, and emotional impact. The palette system must support smooth gradients, precise LUT behavior, high-impact contrast, wide-gamut color, and subtle transitions across complex visual fields.

That technology opened up a larger product opportunity: what if the same palette intelligence behind Mandelbrot Metal became a dedicated tool for building production-ready color systems?

That is the idea behind **ChromaForge**, a new product from Mandelbrot Metal coming soon.

## Why Build ChromaForge?

Developers and web designers already have many ways to generate attractive swatches. The harder problem is turning those colors into a working interface system. A palette must survive dark mode, high-contrast needs, text readability, component states, focus rings, design-token conventions, and code export requirements.

ChromaForge is built for the practical middle ground between visual exploration and implementation. It keeps the creative feel of color experimentation while adding the checks, previews, token layers, and exports that production teams need.

## Who It Is For

- Developers who need app-ready tokens instead of manually translating swatches into code.
- Web designers who want to test color decisions against real interface states before handoff.
- App builders who need SwiftUI-ready colors, theme variants, and accessible component previews.
- Design-system teams that need primitive, semantic, and component-level tokens with reliable exports.
- Technical founders and indie developers who want a fast way to turn brand colors into production-ready themes.

## What ChromaForge Will Include

Area	Capabilities
<b>Palette authoring</b>	Palette browser with favorites/search, large gradient preview, exact-LUT preview, OKLCH/perceptual generation, locked stops, stop editing, Display-P3/sRGB controls, step markers, dithering preview, and editable notes.
<b>Imports</b>	Photo sampling through PhotosPicker and website palette import from HTML, inline styles, and linked CSS. The importer extracts hex, RGB/RGBA, and HSL/HSLA colors, then ranks likely brand and theme colors.
<b>Developer handoff</b>	Light, dark, and high-contrast theme matrix; primitive tokens; semantic role tokens; component tokens; and one-click AA contrast repair.
<b>Preview and accessibility</b>	Context preview, component preview lab, WCAG contrast inspection, semantic theme checks, and color-vision simulation for protanopia, deuteranopia, and tritanopia.
<b>Exports</b>	JSON, raw CSS variables, Semantic CSS, Tailwind, SwiftUI Color extensions, Tokens Studio JSON, Style Dictionary JSON, and CSV.

## How ChromaForge Helps Developers

For developers, ChromaForge reduces the time between "these colors look good" and "these colors work in the app." It can generate semantic roles such as background, foreground, surface, primary, accent, success, warning, and danger, then preview those colors in buttons, forms, alerts, cards, focus rings, and code surfaces.

The Developer Handoff workflow is intended to produce practical assets: Semantic CSS with light, dark, and high-contrast themes; Tailwind-ready color structures; SwiftUI Color extensions; Tokens Studio JSON; Style Dictionary JSON; and CSV for additional workflows.

## How ChromaForge Helps Web Designers

For web designers, ChromaForge makes color decisions easier to validate before they become design debt. A palette may look beautiful as swatches but fail when applied to text, disabled states, alerts, buttons, or focus rings. ChromaForge gives designers a way to test real interface scenarios earlier and collaborate with developers using shared token language.

Website import creates a useful bridge from an existing brand presence to a working color system. Designers can start from a live site, extract likely brand and theme colors, then refine them into accessible, exportable themes.

## Planned Launch Pricing

ChromaForge is planned as a freemium product: free to explore, Pro for production-ready color systems.

Plan	Price	Cadence	What It Unlocks
<b>Free</b>	\$0	Forever	5 custom/imported palettes, 2 website imports, photo import, basic contrast checks, JSON and raw CSS exports.
<b>Pro Annual</b>	\$19.99	Per year launch price	Unlimited palettes and website imports, theme matrix, token layers, AA repair, component previews, and all advanced exports. Planned to move to \$29.99/year after launch validation.
<b>Pro Monthly</b>	\$2.99	Per month	Same Pro feature access with flexible monthly billing for short projects or occasional use.
<b>Founder Lifetime</b>	\$49.99	One time	All Pro features for early supporters. Planned to move to \$69.99-\$79.99 after the founder period.

## The Direction

ChromaForge is not meant to be another casual palette generator. It is a palette-to-product tool: a focused environment for importing, generating, testing, repairing, previewing, and exporting color systems.

The product is still in development, but the direction is clear: take the color technology and visual experimentation DNA of Mandelbrot Metal, then focus it into a practical tool for people building websites, apps, and design systems across iPhone, iPad, and Mac.

**ChromaForge. From beautiful palettes to production-ready color systems.**

# Sample Screenshots

