

Design Skills

- [/canvas-design](#)
- [/frontend-design](#)
- [/figma-design](#)
- [/design-system](#)
- [/brand-guidelines](#)

/canvas-design

Source: `~/ .claude/skills/canvas-design/SKILL.md`

name: canvas-design **description:**
Create beautiful visual art in .png and .pdf documents using design philosophy. You should use this skill when the user asks to create a poster, piece of art, design, or other static piece. Create original visual designs, never copying existing artists' work to avoid copyright violations. **license:** Complete terms in LICENSE.txt

These are instructions for creating design philosophies - aesthetic movements that are then EXPRESSED VISUALLY. Output only .md files, .pdf files, and .png files.

Complete this in two steps:

1. Design Philosophy Creation (.md file)
2. Express by creating it on a canvas (.pdf file or .png file)

First, undertake this task:

DESIGN PHILOSOPHY CREATION

To begin, create a VISUAL PHILOSOPHY (not layouts or templates) that will be interpreted through:

- Form, space, color, composition
- Images, graphics, shapes, patterns
- Minimal text as visual accent

THE CRITICAL UNDERSTANDING

- What is received: Some subtle input or instructions by the user that should be taken into account, but used as a foundation; it should not constrain creative freedom.
- What is created: A design philosophy/aesthetic movement.
- What happens next: Then, the same version receives the philosophy and EXPRESSES IT VISUALLY - creating artifacts that are 90% visual design, 10% essential text.

Consider this approach:

- Write a manifesto for an art movement
- The next phase involves making the artwork

The philosophy must emphasize: Visual expression. Spatial communication. Artistic interpretation. Minimal words.

HOW TO GENERATE A VISUAL PHILOSOPHY

Name the movement (1-2 words): "Brutalist Joy" / "Chromatic Silence" / "Metabolist Dreams"

Articulate the philosophy (4-6 paragraphs - concise but complete):

To capture the VISUAL essence, express how the philosophy manifests through:

- Space and form
- Color and material
- Scale and rhythm
- Composition and balance
- Visual hierarchy

CRITICAL GUIDELINES:

- **Avoid redundancy:** Each design aspect should be mentioned once. Avoid repeating points about color theory, spatial relationships, or typographic principles unless adding new depth.

- **Emphasize craftsmanship REPEATEDLY:** The philosophy MUST stress multiple times that the final work should appear as though it took countless hours to create, was labored over with care, and comes from someone at the absolute top of their field. This framing is essential - repeat phrases like "meticulously crafted," "the product of deep expertise," "painstaking attention," "master-level execution."
- **Leave creative space:** Remain specific about the aesthetic direction, but concise enough that the next Claude has room to make interpretive choices also at a extremely high level of craftsmanship.

The philosophy must guide the next version to express ideas VISUALLY, not through text. Information lives in design, not paragraphs.

PHILOSOPHY EXAMPLES

"Concrete Poetry" Philosophy: Communication through monumental form and bold geometry. Visual expression: Massive color blocks, sculptural typography (huge single words, tiny labels), Brutalist spatial divisions, Polish poster energy meets Le Corbusier. Ideas expressed through visual weight and spatial tension, not explanation. Text as rare, powerful gesture - never paragraphs, only essential words integrated into the visual architecture. Every element placed with the precision of a master craftsman.

"Chromatic Language" Philosophy: Color as the primary information system. Visual expression: Geometric precision where color zones create meaning. Typography minimal - small sans-serif labels letting chromatic fields communicate. Think Josef Albers' interaction meets data visualization. Information encoded spatially and chromatically. Words only to anchor what color already shows. The result of painstaking chromatic calibration.

"Analog Meditation" Philosophy: Quiet visual contemplation through texture and breathing room. Visual expression: Paper grain, ink bleeds, vast negative space. Photography and illustration dominate. Typography whispered (small, restrained, serving the visual). Japanese photobook aesthetic. Images breathe across pages. Text appears sparingly - short phrases, never explanatory blocks. Each composition balanced with the care of a meditation practice.

"Organic Systems" Philosophy: Natural clustering and modular growth patterns. Visual expression: Rounded forms, organic arrangements, color from nature through architecture. Information shown through visual diagrams, spatial relationships, iconography. Text only for key labels floating in space. The composition tells the story through expert spatial orchestration.

"Geometric Silence" Philosophy: Pure order and restraint. Visual expression: Grid-based precision, bold photography or stark graphics, dramatic negative space. Typography precise but minimal - small essential text, large quiet zones. Swiss formalism meets Brutalist material honesty. Structure communicates, not words. Every alignment the work of countless refinements.

These are condensed examples. The actual design philosophy should be 4-6 substantial paragraphs.

ESSENTIAL PRINCIPLES

- **VISUAL PHILOSOPHY:** Create an aesthetic worldview to be expressed through design
- **MINIMAL TEXT:** Always emphasize that text is sparse, essential-only, integrated as visual element - never lengthy
- **SPATIAL EXPRESSION:** Ideas communicate through space, form, color, composition - not paragraphs
- **ARTISTIC FREEDOM:** The next Claude interprets the philosophy visually - provide creative room
- **PURE DESIGN:** This is about making ART OBJECTS, not documents with decoration
- **EXPERT CRAFTSMANSHIP:** Repeatedly emphasize the final work must look meticulously crafted, labored over with care, the product of countless hours by someone at the top of their field

The design philosophy should be 4-6 paragraphs long. Fill it with poetic design philosophy that brings together the core vision. Avoid repeating the same points. Keep the design philosophy generic without mentioning the intention of the art, as if it can be used wherever. Output the design philosophy as a .md file.

DEDUCING THE SUBTLE REFERENCE

CRITICAL STEP: Before creating the canvas, identify the subtle conceptual thread from the original request.

THE ESSENTIAL PRINCIPLE: The topic is a **subtle, niche reference embedded within the art itself** - not always literal, always sophisticated. Someone familiar with the subject should feel it intuitively, while others simply experience a masterful abstract composition. The design philosophy provides the aesthetic language. The deduced topic provides the soul - the quiet conceptual DNA woven invisibly into form, color, and composition.

This is **VERY IMPORTANT:** The reference must be refined so it enhances the work's depth without announcing itself. Think like a jazz musician quoting another song - only those who know will catch it, but everyone appreciates the music.

CANVAS CREATION

With both the philosophy and the conceptual framework established, express it on a canvas. Take a moment to gather thoughts and clear the mind. Use the design philosophy created and the

instructions below to craft a masterpiece, embodying all aspects of the philosophy with expert craftsmanship.

IMPORTANT: For any type of content, even if the user requests something for a movie/game/book, the approach should still be sophisticated. Never lose sight of the idea that this should be art, not something that's cartoony or amateur.

To create museum or magazine quality work, use the design philosophy as the foundation. Create one single page, highly visual, design-forward PDF or PNG output (unless asked for more pages). Generally use repeating patterns and perfect shapes. Treat the abstract philosophical design as if it were a scientific bible, borrowing the visual language of systematic observation—dense accumulation of marks, repeated elements, or layered patterns that build meaning through patient repetition and reward sustained viewing. Add sparse, clinical typography and systematic reference markers that suggest this could be a diagram from an imaginary discipline, treating the invisible subject with the same reverence typically reserved for documenting observable phenomena. Anchor the piece with simple phrase(s) or details positioned subtly, using a limited color palette that feels intentional and cohesive. Embrace the paradox of using analytical visual language to express ideas about human experience: the result should feel like an artifact that proves something ephemeral can be studied, mapped, and understood through careful attention. This is true art.

Text as a contextual element: Text is always minimal and visual-first, but let context guide whether that means whisper-quiet labels or bold typographic gestures. A punk venue poster might have larger, more aggressive type than a minimalist ceramics studio identity. Most of the time, font should be thin. All use of fonts must be design-forward and prioritize visual communication. Regardless of text scale, nothing falls off the page and nothing overlaps. Every element must be contained within the canvas boundaries with proper margins. Check carefully that all text, graphics, and visual elements have breathing room and clear separation. This is non-negotiable for professional execution. **IMPORTANT: Use different fonts if writing text. Search the `./canvas-fonts` directory. Regardless of approach, sophistication is non-negotiable.**

Download and use whatever fonts are needed to make this a reality. Get creative by making the typography actually part of the art itself -- if the art is abstract, bring the font onto the canvas, not typeset digitally.

To push boundaries, follow design instinct/intuition while using the philosophy as a guiding principle. Embrace ultimate design freedom and choice. Push aesthetics and design to the frontier.

CRITICAL: To achieve human-crafted quality (not AI-generated), create work that looks like it took countless hours. Make it appear as though someone at the absolute top of their field labored over every detail with painstaking care. Ensure the composition, spacing, color choices, typography - everything screams expert-level craftsmanship. Double-check that nothing overlaps, formatting is flawless, every detail perfect. Create something that could be shown to people to prove expertise and rank as undeniably impressive.

Output the final result as a single, downloadable .pdf or .png file, alongside the design philosophy used as a .md file.

FINAL STEP

IMPORTANT: The user ALREADY said "It isn't perfect enough. It must be pristine, a masterpiece if craftsmanship, as if it were about to be displayed in a museum."

CRITICAL: To refine the work, avoid adding more graphics; instead refine what has been created and make it extremely crisp, respecting the design philosophy and the principles of minimalism entirely. Rather than adding a fun filter or refactoring a font, consider how to make the existing composition more cohesive with the art. If the instinct is to call a new function or draw a new shape, STOP and instead ask: "How can I make what's already here more of a piece of art?"

Take a second pass. Go back to the code and refine/polish further to make this a philosophically designed masterpiece.

MULTI-PAGE OPTION

To create additional pages when requested, create more creative pages along the same lines as the design philosophy but distinctly different as well. Bundle those pages in the same .pdf or many .pngs. Treat the first page as just a single page in a whole coffee table book waiting to be filled. Make the next pages unique twists and memories of the original. Have them almost tell a story in a very tasteful way. Exercise full creative freedom.

/frontend-design

Source: `~/ .claude/skills/frontend-design/SKILL.md`

name: frontend-design **description:**
Create distinctive, production-grade frontend interfaces with high design quality. Use this skill when the user asks to build web components, pages, artifacts, posters, or applications (examples include websites, landing pages, dashboards, React components, HTML/CSS layouts, or when styling/beautifying any web UI). Generates creative, polished code and UI design that avoids generic AI aesthetics. **license:** Complete terms in LICENSE.txt

This skill guides creation of distinctive, production-grade frontend interfaces that avoid generic "AI slop" aesthetics. Implement real working code with exceptional attention to aesthetic details and creative choices.

The user provides frontend requirements: a component, page, application, or interface to build. They may include context about the purpose, audience, or technical constraints.

Design Thinking

Before coding, understand the context and commit to a BOLD aesthetic direction:

- **Purpose:** What problem does this interface solve? Who uses it?
- **Tone:** Pick an extreme: brutally minimal, maximalist chaos, retro-futuristic, organic/natural, luxury/refined, playful/toy-like, editorial/magazine, brutalist/raw, art deco/geometric, soft/pastel, industrial/utilitarian, etc. There are so many flavors to choose from. Use these for inspiration but design one that is true to the aesthetic direction.
- **Constraints:** Technical requirements (framework, performance, accessibility).
- **Differentiation:** What makes this UNFORGETTABLE? What's the one thing someone will remember?

CRITICAL: Choose a clear conceptual direction and execute it with precision. Bold maximalism and refined minimalism both work - the key is intentionality, not intensity.

Then implement working code (HTML/CSS/JS, React, Vue, etc.) that is:

- Production-grade and functional
- Visually striking and memorable
- Cohesive with a clear aesthetic point-of-view
- Meticulously refined in every detail

Frontend Aesthetics Guidelines

Focus on:

- **Typography:** Choose fonts that are beautiful, unique, and interesting. Avoid generic fonts like Arial and Inter; opt instead for distinctive choices that elevate the frontend's aesthetics; unexpected, characterful font choices. Pair a distinctive display font with a refined body font.
- **Color & Theme:** Commit to a cohesive aesthetic. Use CSS variables for consistency. Dominant colors with sharp accents outperform timid, evenly-distributed palettes.
- **Motion:** Use animations for effects and micro-interactions. Prioritize CSS-only solutions for HTML. Use Motion library for React when available. Focus on high-impact moments: one well-orchestrated page load with staggered reveals (animation-delay) creates more

delight than scattered micro-interactions. Use scroll-triggering and hover states that surprise.

- **Spatial Composition:** Unexpected layouts. Asymmetry. Overlap. Diagonal flow. Grid-breaking elements. Generous negative space OR controlled density.
- **Backgrounds & Visual Details:** Create atmosphere and depth rather than defaulting to solid colors. Add contextual effects and textures that match the overall aesthetic. Apply creative forms like gradient meshes, noise textures, geometric patterns, layered transparencies, dramatic shadows, decorative borders, custom cursors, and grain overlays.

NEVER use generic AI-generated aesthetics like overused font families (Inter, Roboto, Arial, system fonts), cliched color schemes (particularly purple gradients on white backgrounds), predictable layouts and component patterns, and cookie-cutter design that lacks context-specific character.

Interpret creatively and make unexpected choices that feel genuinely designed for the context. No design should be the same. Vary between light and dark themes, different fonts, different aesthetics. NEVER converge on common choices (Space Grotesk, for example) across generations.

IMPORTANT: Match implementation complexity to the aesthetic vision. Maximalist designs need elaborate code with extensive animations and effects. Minimalist or refined designs need restraint, precision, and careful attention to spacing, typography, and subtle details. Elegance comes from executing the vision well.

Remember: Claude is capable of extraordinary creative work. Don't hold back, show what can truly be created when thinking outside the box and committing fully to a distinctive vision.

/figma-design

Source: `~/ .claude/skills/figma-design/SKILL.md`

name: figma-design description: "Figma specialist for EXISTING Figma files. Extracts tokens, generates React code, validates builds, exports assets. Use when working WITH a Figma file. For NEW designs from scratch, use /design-system." argument-hint: "[Figma file key or 'drop' shortcut] [what to do — extract, generate, validate, export]"

Figma Design Specialist

You work with EXISTING Figma files — extracting tokens, generating code, validating builds, and exporting assets.

Knowledge Base: `~/system/context/figma-knowledge-base.md` — READ THIS FIRST for all Figma operations.

When This Skill Triggers

- "extract from Figma", "Figma tokens", "export from Figma"
- "generate code from Figma", "Figma to React"
- "compare with Figma", "validate against Figma"
- Working with an existing Figma file
- Design token extraction or sync

Tools Available

Tool	Command	What It Does
<code>figma-extract.js</code>	<code>extract-tokens</code> , <code>extract-components</code> , <code>frame-to-prompt</code> , <code>export-image</code> , <code>list-nodes</code>	Read Figma REST API
<code>figma-to-react.js</code>	<code><file-key> <node-id></code>	Generate React + Tailwind from Figma frame
<code>figma-token-sync.js</code>	<code><file-key> --format all</code>	Variables → W3C tokens + Tailwind + CSS vars
<code>figma-validate.js</code>	<code>compare <file-key> <node-id> <url></code>	Visual diff: Figma vs built page
<code>design-to-code.js</code>	<code>assemble --stitch-code X --assets-dir Y</code>	Stitch HTML → React TSX

Common Workflows

Extract Design Tokens

```
# Get all tokens (colors, typography, spacing)
node ~/system/tools/figma-extract.js extract-tokens <file-key>

# Sync variables to code-ready formats
node ~/system/tools/figma-token-sync.js <file-key> --format all --output ./tokens/
```

Generate React from Figma Frame

```
# List nodes to find the right frame
node ~/system/tools/figma-extract.js list-nodes <file-key>
```

```
# Generate React + Tailwind component
node ~/system/tools/figma-to-react.js <file-key> <node-id> --output
./src/components/Screen.tsx
```

Export Assets

```
# Export frame as PNG (2x retina)
node ~/system/tools/figma-extract.js export-image <file-key> <node-id> --format png --scale 2
--output ./public/frame.png

# Export as SVG (icons, logos)
node ~/system/tools/figma-extract.js export-image <file-key> <node-id> --format svg --output
./public/icon.svg
```

Validate Build Against Design

```
# Compare Figma design vs built page – outputs diff report
node ~/system/tools/figma-validate.js compare <file-key> <node-id> http://localhost:3000/login
```

Drop App Shortcuts

File key: `P535qC6nAREfoTsMWf0qqi`

Page	Node ID	Description
Page 1	<code>0:1</code>	Main page
Design System	<code>6:2</code>	Colors, typography, components
Screen Test	<code>6:140</code>	Component testing
Screens	<code>6:142</code>	Full app screens
Login v2 UX	<code>6:175</code>	Login flow variations

Figma REST API Quick Reference

Endpoint	What
<code>GET /v1/files/{key}</code>	Full file data

Endpoint	What
GET /v1/files/{key}/nodes?ids=X	Specific nodes
GET /v1/images/{key}?ids=X&format=png&scale=2	Export as image
GET /v1/files/{key}/variables/local	Design variables
GET /v1/files/{key}/components	Components

Auth: X-Figma-Token header from ~/system/config/figma.json **Rate limits:** 6-20 req/min (Tier 1), use 200ms delay between requests **Export:** PNG/JPG up to 4x scale (max 32MP), SVG/PDF at 1x only, URLs expire in 30 days

Figma Live Bridge (WebSocket — PROTOTYPE)

For direct manipulation of Figma Desktop. **Reliability:** ~40% — use REST API tools above instead when possible.

Setup: Figma Desktop → Plugins → Development → Claude MCP Plugin → get channel ID

Connection: WebSocket on port 3055 via bun socket **Commands:** create_frame, create_text, create_rectangle, create_ellipse, set_fill_color, get_document_info, get_node_info **Timeouts:** set_corner_radius, set_effects, set_auto_layout — use workarounds

Design Token Architecture (3-Tier)

PRIMITIVE → raw values (blue-500, spacing-16)
 SEMANTIC → purpose (color-primary → blue-500)
 COMPONENT → scoped (button-bg → color-primary)

Naming: {category}-{role}-{modifier}-{state} (kebab-case) **Modes:** Light/Dark via Figma Variable modes

Visual Verification (ZAKON #0.1)

ALWAYS validate builds against Figma. Run figma-validate.js before claiming "done". List DIFFERENCES, not similarities. If you can't find any differences, you're not looking carefully enough.

Quality Checklist

- Tokens extracted from Figma (not guessed)
- Colors match Figma exactly (verified with hex comparison)
- Typography matches (font family, size, weight, line-height)
- Spacing matches (padding, gaps, margins)
- Assets exported from Figma (not hand-drawn SVG)
- Visual validation run (`figma-validate.js`)
- Difference percentage below 10%

Reference

- **Knowledge base:** `~/system/context/figma-knowledge-base.md`
- **REST API reference:** `~/system/context/figma-rest-api-reference-2025-2026.md`
- **Config:** `~/system/config/figma.json`
- **Manifest:** `~/system/tools/manifest.md`

/design-system

Source: `~/ .claude/skills/design-system/SKILL.md`

name: design-system description: "End-to-end design-to-build pipeline for NEW projects. Stitch generates designs ? Figma import ? extract tokens ? generate code ? validate. Use for new designs from scratch." argument-hint: "[project brief — app name, industry, brand colors, screens to design]"

Design-to-Build Pipeline

Create new designs from scratch and build them into production code. For working with EXISTING Figma files, use `/figma-design` instead.

Knowledge Base: `~/system/context/figma-knowledge-base.md` — READ THIS FIRST.

The Pipeline (7 Steps)

BRIEF → STITCH → FIGMA → EXTRACT → BUILD → VALIDATE → DEPLOY

↓

↓

↓

↓

↓

↓

↓

Spec Generate Import Tokens React Compare Ship
(FREE) (manual) (auto) (auto) (visual)

Step 1: BRIEF — Parse Requirements

Extract from request:

- **App name** and industry (fintech, SaaS, music, consulting)
- **Brand** — primary color, accent, background
- **Screens** to design (login, dashboard, send-money, etc.)
- **Elements** — what must be on screen
- **Audience** — who uses this
- **Vibe** — 3 keywords (e.g., "trustworthy, effortless, Scandinavian")

Step 2: STITCH — Generate Design (FREE)

```
node ~/system/tools/stitch-generate.js \  
  --brief "[APP]" --screen "[SCREEN]" --industry "[INDUSTRY]" \  
  --primary "[HEX]" --secondary "[HEX]" \  
  --vibe "[KW1], [KW2], [KW3]" \  
  --elements "[EL1],[EL2],[EL3]" \  
  --model pro --options 3
```

Output: 3 style variants in `~/system/design-output/stitch-<timestamp>/`

Present for approval:

```
node ~/system/tools/design-board.js create "[Project] [Screen]" "reviewer@email" \  
  --options '[...png paths...]' --recommend 2
```

WAIT for CEO/client approval before proceeding.

Step 3: FIGMA — Import Approved Design (Manual Step)

Method A: In Stitch, "Copy to Figma" → Cmd+V in Figma Desktop **Method B:** Download HTML/CSS from Stitch → use `html.to.design` plugin in Figma (80-90% accuracy) **Method C:** Use `figma-populate.js` WebSocket bridge (unreliable — last resort)

After import: **Figma IS the source of truth.** All subsequent work reads FROM Figma.

Step 4: EXTRACT — Tokens + Assets from Figma

```
# Design tokens → all formats
node ~/system/tools/figma-token-sync.js <file-key> --format all --output ./tokens/

# Individual assets (logos, icons)
node ~/system/tools/figma-extract.js export-image <file-key> <node-id> --format svg --output
./public/logo.svg

# Generate implementation prompt
node ~/system/tools/figma-extract.js frame-to-prompt <file-key> <node-id>
```

Step 5: BUILD — Code from Figma Data

Option A: Direct Figma → React (NEW)

```
node ~/system/tools/figma-to-react.js <file-key> <node-id> --output ./src/app/page.tsx
```

Option B: Stitch HTML → React (existing)

```
node ~/system/tools/design-to-code.js assemble \
  --stitch-code code.html --assets-dir exports/ --target-page page.tsx --preserve-logic
```

Rules:

1. Logo/icons → exported from Figma, NEVER hand-drawn SVG
2. Colors → extracted token values, not guessed hex codes
3. Typography → match extracted font/size/weight exactly
4. Spacing → match extracted spacing values
5. Layout → follow Auto Layout → Flexbox mapping

Step 6: VALIDATE — Compare Code to Design

```
node ~/system/tools/figma-validate.js compare <file-key> <node-id> http://localhost:3000/page
```

ZAKON #0.1: List DIFFERENCES, not similarities. If diff > 10%, fix before proceeding.

Step 7: DEPLOY

```
# Docker build for Fly.io
docker build -t app .
flyctl deploy

# Or Vercel
vercel --prod
```

Design System Architecture

Token Structure (3-Tier MANDATORY)

```
PRIMITIVE: blue-500, gray-900, spacing-16
  ↓
SEMANTIC: color-primary, text-primary, spacing-md
  ↓
COMPONENT: button-bg-primary, card-padding, input-border-focus
```

Typography Scale (Major Third 1.25)

Size	Use	Tailwind
12px	Caption	<code>text-xs</code>
14px	Small body	<code>text-sm</code>
16px	Body (BASE)	<code>text-base</code>
20px	Subtitle	<code>text-xl</code>
25px	H3	<code>text-2xl</code>
31px	H2	<code>text-3xl</code>
39px	H1	<code>text-4xl</code>

Spacing Grid (4px baseline + 8pt elements)

Value	Token	Tailwind	Use
4px	spacing-xs	<code>p-1</code>	Micro spacing

Value	Token	Tailwind	Use
8px	spacing-sm	p-2	Between label↔input
12px	spacing-md-sm	p-3	Compact padding
16px	spacing-md	p-4	Form fields
24px	spacing-lg	p-6	Card padding
32px	spacing-xl	p-8	Between sections
48px	spacing-2xl	p-12	Major breaks
64px	spacing-3xl	p-16	Hero spacing

Color System

Semantic tokens (minimum set):

- color-primary, color-secondary, color-accent
- color-success, color-warning, color-error, color-info
- text-primary, text-secondary, text-disabled
- surface-base, surface-raised, surface-overlay
- border-default, border-focus, border-error

WCAG: AA minimum (4.5:1 normal text, 3:1 large text)

Elevation (5 levels)

Level	Use	CSS
0	Flat	none
1	Cards	shadow-sm
2	Buttons	shadow-md
3	Dropdowns	shadow-lg
4	Modals	shadow-xl

Industry Patterns

Fintech

- Trust signals: green, navy, white
- BankID/Vipps prominence, security cues

- Biometric-first auth, generous whitespace
- Clear number formatting, transaction lists

SaaS

- Clean layouts, data visualization
- System fonts, minimal palette
- Dashboard patterns, data tables

Music/Creative

- Dark themes, neon accents
- Bold typography, gradient meshes
- Dynamic visuals, high contrast

Professional Services

- Minimal, structured layouts
- Restrained palette, strong hierarchy
- Corporate, competent, reliable

Quality Gate (MANDATORY before delivery)

- Design approved by CEO/client (Step 3 gate)
- Tokens extracted from Figma (NOT guessed)
- `figma-validate.js` run — diff < 10%
- DIFFERENCES listed explicitly
- All assets exported from Figma
- WCAG AA contrast verified
- Mobile touch targets \geq 44px

Reference

- **Knowledge base:** `~/system/context/figma-knowledge-base.md`
- **REST API:** `~/system/context/figma-rest-api-reference-2025-2026.md`
- **Config:** `~/system/config/figma.json`

- **Tools:** `~/system/tools/manifest.md`

/brand-guidelines

Source: `~/ .claude/skills/brand-guidelines/SKILL.md`

name: brand-guidelines **description:** Applies Anthropic's official brand colors and typography to any sort of artifact that may benefit from having Anthropic's look-and-feel. Use it when brand colors or style guidelines, visual formatting, or company design standards apply. **license:** Complete terms in LICENSE.txt

Anthropic Brand Styling

Overview

To access Anthropic's official brand identity and style resources, use this skill.

Keywords: branding, corporate identity, visual identity, post-processing, styling, brand colors, typography, Anthropic brand, visual formatting, visual design

Brand Guidelines

Colors

Main Colors:

- Dark: #141413 - Primary text and dark backgrounds
- Light: #faf9f5 - Light backgrounds and text on dark
- Mid Gray: #b0aea5 - Secondary elements
- Light Gray: #e8e6dc - Subtle backgrounds

Accent Colors:

- Orange: #d97757 - Primary accent
- Blue: #6a9bcc - Secondary accent
- Green: #788c5d - Tertiary accent

Typography

- **Headings:** Poppins (with Arial fallback)
- **Body Text:** Lora (with Georgia fallback)
- **Note:** Fonts should be pre-installed in your environment for best results

Features

Smart Font Application

- Applies Poppins font to headings (24pt and larger)
- Applies Lora font to body text
- Automatically falls back to Arial/Georgia if custom fonts unavailable
- Preserves readability across all systems

Text Styling

- Headings (24pt+): Poppins font
- Body text: Lora font
- Smart color selection based on background
- Preserves text hierarchy and formatting

Shape and Accent Colors

- Non-text shapes use accent colors
- Cycles through orange, blue, and green accents
- Maintains visual interest while staying on-brand

Technical Details

Font Management

- Uses system-installed Poppins and Lora fonts when available
- Provides automatic fallback to Arial (headings) and Georgia (body)
- No font installation required - works with existing system fonts
- For best results, pre-install Poppins and Lora fonts in your environment

Color Application

- Uses RGB color values for precise brand matching
- Applied via python-pptx's RGBColor class
- Maintains color fidelity across different systems