

– REFERENCE –

The Design *Lexicon*

*A vocabulary for prompting better UI.
Specific words produce specific outcomes.*

01	LAYOUT	STRUCTURE & COMPOSITION
02	COMPONENTS	THE BUILDING BLOCKS
03	STATES	STATIC LOOKS VS REAL UX
04	SPACING	THE BIG ONE
05	ALIGNMENT	LINING THINGS UP
06	FORM FACTOR	RESPONSIVE DESIGN
07	TYPOGRAPHY	THE VOICE ON SCREEN
08	FONTS	VOICE & WEIGHT
09	COLOR	HUE & HIERARCHY
10	FIELD GUIDE	VAGUE → SPECIFIC
11	HEURISTICS	FOUR PATTERNS
12	STARTER	FILL IN THE BLANKS

Swap "make it look better" for the language inside, and watch your prompts start landing.

How elements are arranged on the page. The bones of every interface.

Hierarchy

The visual order of importance. Headlines should look like parents. Subheads should look like children.

USE WHEN your H1 and subhead look the same size, nothing pops, the eye doesn't know where to start.

— PROMPT

Establish stronger hierarchy between the H1 and subhead. Make the H1 about 2x bigger and the subhead muted gray.

SIBLINGS

Newsletter for AI writers

Practical playbooks every Tuesday.

Subscribe

PARENT + CHILD

Newsletter

Practical playbooks every Tuesday.

Subscribe

Container

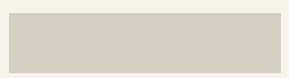
The box your content lives in. Without a max-width, content stretches edge-to-edge on wide screens and stops feeling intentional.

USE WHEN content sprawls across the full screen on desktop and reading feels exhausting.

— PROMPT

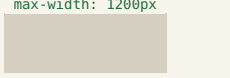
Wrap the main content in a container with max-width: 1200px and center it on the page.

NO MAX-WIDTH



CONSTRAINED

max-width: 1200px



Above the fold


What's visible before the user scrolls. The most valuable real estate on the page — and easy to waste.

USE WHEN important content (hero, value prop, CTA) is hidden below the first screen.

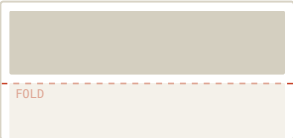
— PROMPT

Move the headline, key benefit, and primary CTA above the fold (first 600px of vertical space).

CTA BELOW



CTA ABOVE



White space

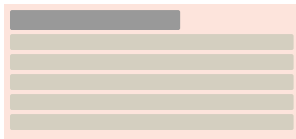
The empty area around and between elements. Carries as much weight as the content itself — not wasted, just earned.

USE WHEN everything is fighting for attention and the page feels visually loud.

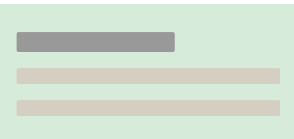
— PROMPT

Add generous white space between sections. Aim for 80px of vertical separation on desktop.

CROWDED



SPACIOUS



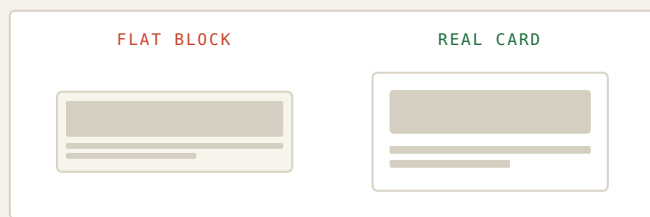
Stop describing what you want. Start naming what it is.

Card

A self-contained block grouping related info — image, title, description, action — with a subtle border or shadow. The most common UI pattern in modern design.

USE WHEN you have repeated items (products, articles, profiles) that should each feel like its own object.

— PROMPT
Display these items as a card grid. Each card has an image, title, two lines of description, and a CTA button.

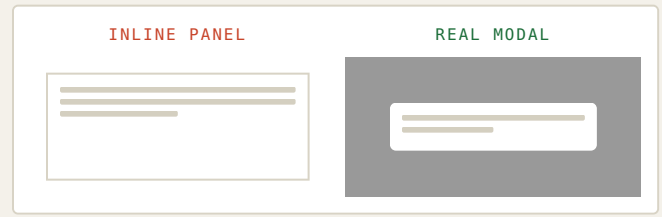


Modal

A focused overlay that interrupts everything else — for confirmations, forms, or important decisions. Background dims to lock focus on the panel.

USE WHEN you need the user to make a single decision before continuing, like confirming a delete or completing a signup.

— PROMPT
Use a modal with a dimmed backdrop, centered panel, clear title, primary and secondary buttons, and a close affordance.

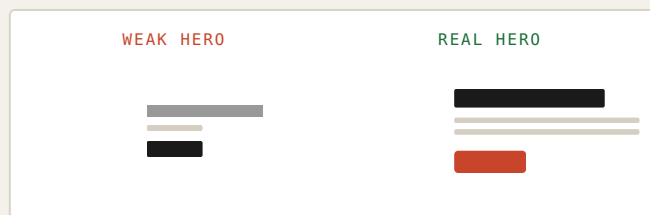


Hero section

The dominant top-of-page block: big headline, supporting line, primary CTA. The first thing visitors see and the most valuable real estate on any page.

USE WHEN your landing page opens with a small headline that doesn't grab attention or sell the value prop.

— PROMPT
Build a hero section above the fold with a 48px headline, supporting paragraph, and a single accent-colored CTA button.

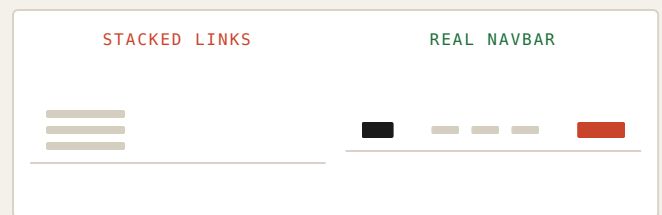


Navbar

The top-of-page navigation: logo on the left, links in the middle or right, primary CTA at the far right. The wayfinding system for everything else.

USE WHEN navigation is a vertical list, links float without anchor points, or there's no clear primary action.

— PROMPT
Build a sticky navbar: logo on the left, 3–4 navigation links centered or right, primary CTA in the accent color at the far right.



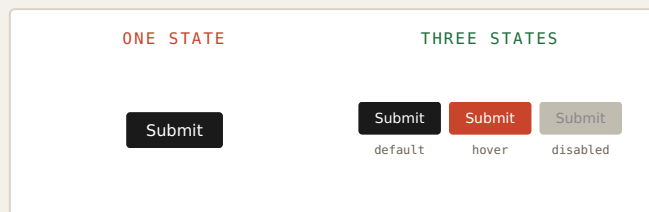
A button isn't a button. It's five buttons stacked on top of each other, each visible at a different moment.

Button states

Default, hover, active, focus, and disabled. Each looks slightly different so users know it's interactive and what it'll do next. Skipping these is the loudest "AI-generated UI" tell.

USE WHEN buttons feel flat or dead — nothing happens on hover, no visual change on click, no signal that they're interactive.

— PROMPT
Design all five button states: default, hover (darker), active (pressed), focus (ring), and disabled (muted, no cursor pointer).

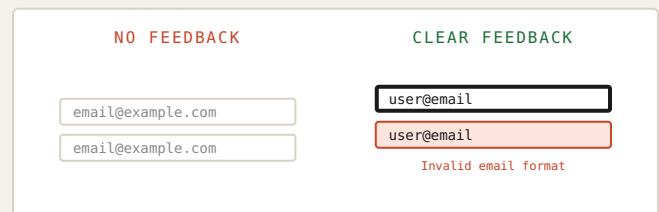


Form states

Inputs need empty, focused, filled, error, and success states. Each state tells the user where they are and what's working or broken — without it, forms feel hostile.

USE WHEN your form looks identical whether the user has typed something correct or made a mistake.

— PROMPT
Design four form states: empty (placeholder), focused (border highlight), error (red border + message below), and success (subtle green check).

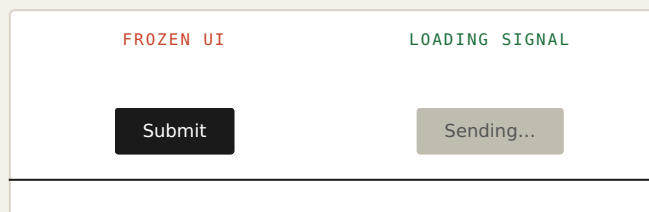


Loading state

What the user sees while something is happening. Spinners, skeleton screens, disabled buttons with "Sending..." labels. Without it, users tap the same button five times.

USE WHEN actions take more than half a second and the UI gives no signal that anything's happening.

— PROMPT
Add a loading state to the submit button. Disable it, change the label to "Sending...", and show a spinner while the request is in flight.

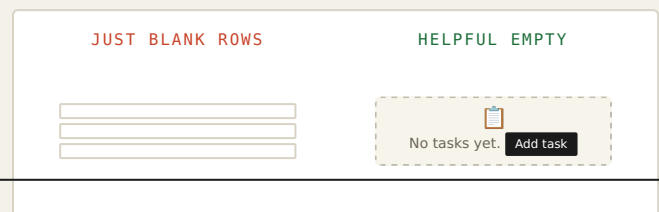


Empty state

What users see when there's no data yet — first-time users, empty inboxes, no search results. A real opportunity, not a void. Should explain what's missing and how to fill it.

USE WHEN a list, table, or page looks broken because there's nothing in it yet.

— PROMPT
Design an empty state with an icon, a one-line explanation of why it's empty, and a primary action to fill it.



When something feels "off," nine times out of ten the answer is here.

Spacing scale

A consistent set of values (4, 8, 16, 24, 32, 48, 64) that every gap, padding, and margin snaps to.

USE WHEN the layout feels random, gaps look inconsistent, nothing lines up.

— PROMPT

Use an 8-point spacing scale throughout. Every margin, padding, and gap should be a multiple of 8.



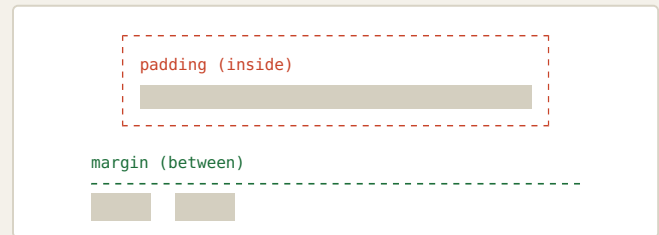
Padding vs margin

Padding is space *inside* an element. Margin is space *outside* it. They fix different problems.

USE WHEN elements look cramped (padding) or stuck together (margin).

— PROMPT

Add 16px of padding inside the card. Add 24px of margin between cards.



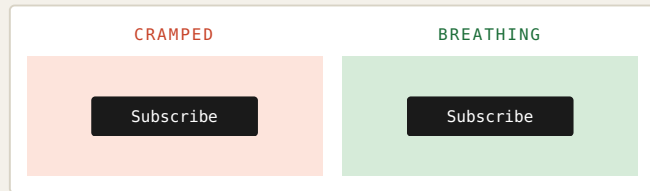
Negative space

Breathing room around primary elements. Empty space carries as much weight as the content does.

USE WHEN the design feels cluttered or the CTA gets visually lost.

— PROMPT

Increase negative space around the primary CTA. Give it room to breathe.



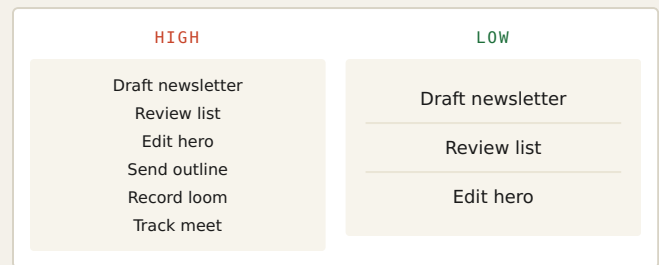
Density

How tightly packed elements are. Reducing density adds breathing room without changing the layout itself.

USE WHEN a list, table, or card feels like a spreadsheet.

— PROMPT

Reduce density on the task list. Increase row padding to 12px and add separators.



Alignment

The invisible discipline. When alignment is right, you don't notice it. When it's wrong, that's all you see.

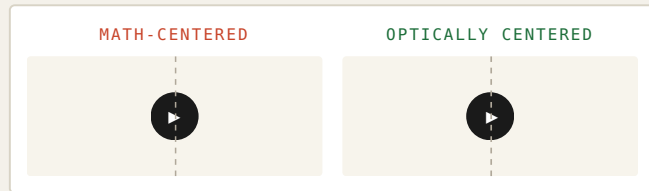
Optical alignment

Adjusting for how the eye perceives shapes, not what the math says. Triangles and play icons need to be nudged off-center to look centered.

USE WHEN an icon or arrow looks slightly off inside its container, even though it's mathematically centered.

— PROMPT

Optically center the play icon inside the button. Nudge it 1–2px to the right of mathematical center.



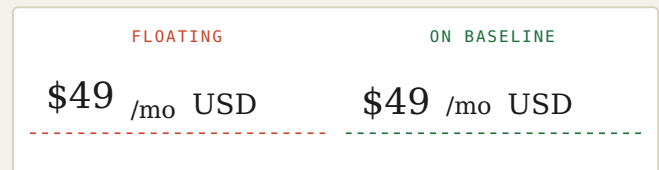
Baseline

The invisible line text sits on. When text of different sizes shares a row, they should all align to it — not their tops or middles.

USE WHEN numbers, labels, or icons next to each other look like they're floating at slightly different heights.

— PROMPT

Align all text in this row to the baseline, not the top or center of the elements.



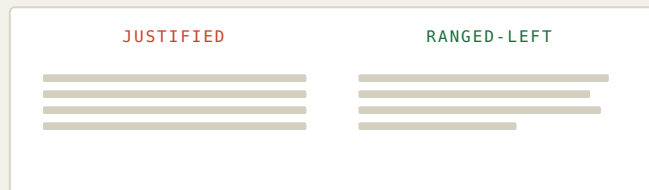
Justified vs ranged

Justified text hits both edges — print-magazine style. Ranged-left (flush left, ragged right) is more natural and reads better on screens.

USE WHEN text justification is creating awkward gaps between words, especially in narrow columns.

— PROMPT

Use ranged-left (flush left, ragged right) alignment for all body copy. Avoid justified text.



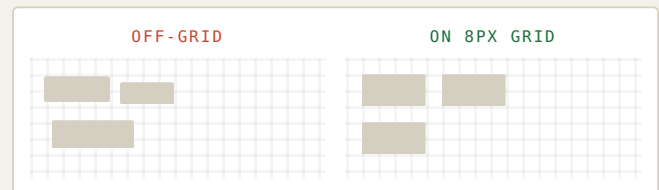
Snap to grid

Forcing every element to align to the same invisible structure. Removes guesswork. Eliminates the dozens of tiny misalignments humans don't catch but feel.

USE WHEN elements look almost-but-not-quite aligned and the layout feels slightly off without you knowing why.

— PROMPT

Snap every element to an 8px grid. All x/y positions and dimensions should be multiples of 8.



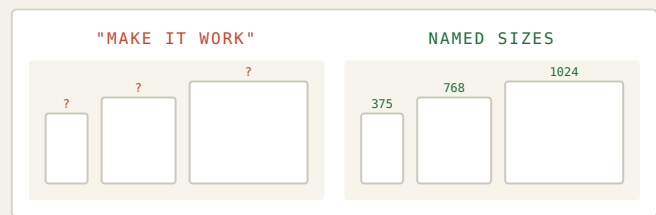
Designs aren't done at one size. They're done at three.

Breakpoints

The exact pixel widths where your layout shifts. Naming them tells Claude how to design for each size, not stretch one design across all of them.

USE WHEN your design looks fine on desktop but cramped on mobile, or vice versa.

— PROMPT
Build responsive designs at three breakpoints: 375px (mobile), 768px (tablet), and 1024px (desktop).

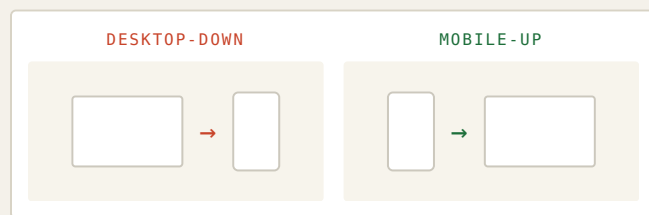


Mobile-first

Designing for the smallest screen first, then scaling up. Forces you to prioritize what's essential before adding for larger screens.

USE WHEN most of your audience is on phones but the design was clearly built for desktop.

— PROMPT
Build this mobile-first. Start at 375px and add complexity as the viewport grows.

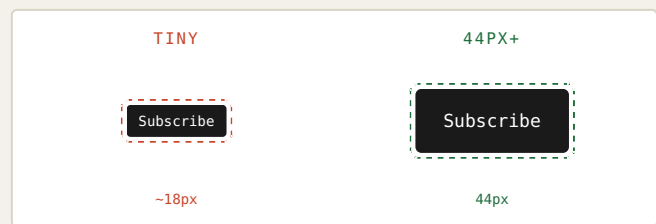


Touch target

The tappable area on mobile. Apple's spec: minimum 44x44px so an average thumb can hit it reliably without missing.

USE WHEN buttons are hard to tap on a phone, users miss small links, or the design was built without thumbs in mind.

— PROMPT
Ensure every tappable element is at least 44x44px on mobile. Add padding to small icons and text links.



Fluid vs fixed

Fluid sizing scales with the viewport using percentages or vw/vh units. Fixed sizing stays the same. Fluid feels modern. Fixed feels rigid.

USE WHEN elements look identical at every screen size — same hero font, same image dimensions, same button padding everywhere.

— PROMPT
Make typography fluid between mobile and desktop using clamp(). Headlines should scale from 32px to 64px.



Typography

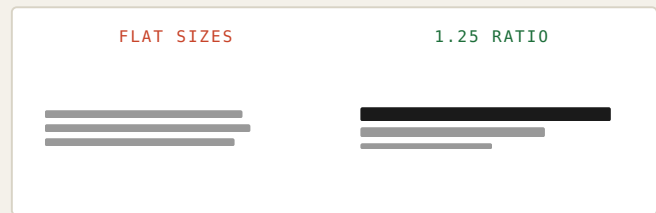
Type is the loudest design choice. Get it right and the rest gets easier.

Type scale

A set of font sizes built on a ratio (1.25 is common). Creates harmony automatically — every level visibly smaller than the one above.

USE WHEN your headlines, subheads, and body text all look like roughly the same size, with no clear visual hierarchy.

- PROMPT
Use a modular type scale with a 1.25 ratio. Build six steps from 14px body up to 48px display.

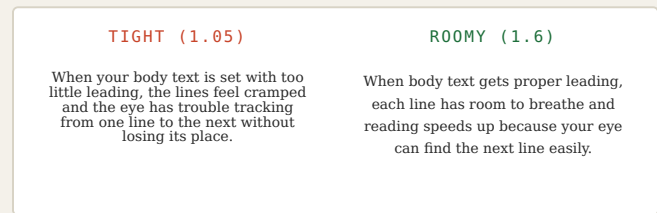


Leading / line-height

The vertical space between lines of text. Body text wants room to breathe (1.5-1.6). Headlines want to feel tight (1.1-1.2).

USE WHEN paragraphs feel cramped or hard to follow, and your eye keeps losing its place.

- PROMPT
Set line-height to 1.6 for body text and 1.15 for headlines.

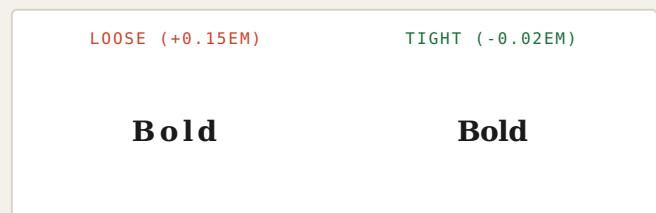


Tracking / letter-spacing

The horizontal space between letters. Big headlines want it tight (negative letter-spacing). Small caps and tiny text want it loose.

USE WHEN large headlines look spaced-out and unfocused, or all-caps text feels mashed together.

- PROMPT
Tighten tracking on the H1 to -0.02em. Loosen tracking on the small uppercase label to 0.15em.

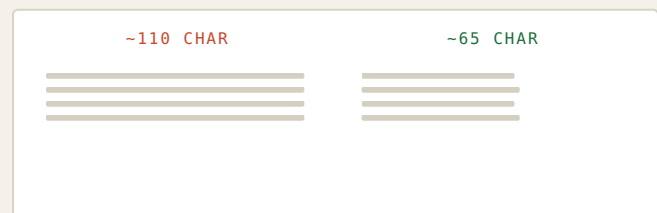


Measure

The width of a column of text. Lines longer than ~75 characters become hard to read because the eye loses its place returning to the next line.

USE WHEN body paragraphs run the full width of a desktop screen and reading feels exhausting.

- PROMPT
Constrain the measure of body text to ~65 characters per line using max-width: 65ch.



Type sets the size and rhythm. Fonts set the voice. Pick wrong and nothing else can save it.

Serif vs sans-serif

Serifs are the small feet on letters — they feel traditional, editorial, authoritative. Sans-serifs ("without serifs") feel modern, clean, neutral. Tell Claude which voice you want.

USE WHEN the design feels generic and could be anywhere — no real personality, no point of view.

— PROMPT
Use a serif for the headline to feel editorial. Use a sans-serif for the body to keep it clean and readable.

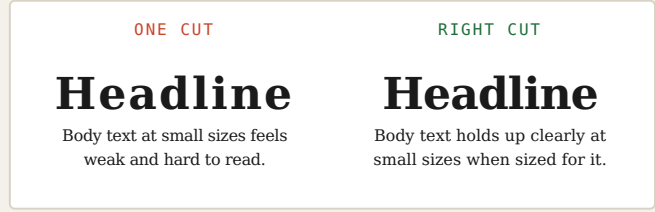


Display vs text

Big fonts and small fonts need different cuts. Display is built for 48px+ headlines (thinner strokes, tighter spacing). Text is built for 12–18px body (thicker strokes, more spacing). Quality fonts ship both.

USE WHEN your headline looks fine but body text feels weak, or your body looks fine but the headline feels heavy.

— PROMPT
Use the Display cut of the font for the H1 (anything above 32px). Use the Text cut for body and small UI labels.

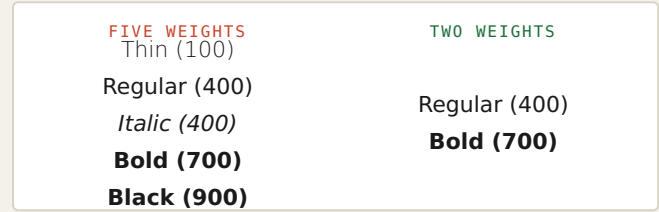


Font weight

How thick the strokes are, on a scale from 100 (Thin) to 900 (Black). Most pro designs use 2–3 weights, max. Using too many is the loudest amateur tell.

USE WHEN the design feels chaotic and every section has a different "look" — usually because it's mixing 5+ weights.

— PROMPT
Use only two font weights: Regular (400) for body text and Bold (700) for emphasis and headlines.



Type pairing

Choosing two fonts that work together. The classic move: pair a serif headline with a sans-serif body. They contrast without fighting.

USE WHEN you want more personality than one font alone, but two similar fonts are blurring together.

— PROMPT
Pair a serif headline (e.g., Fraunces) with a clean sans-serif body (e.g., Inter). Limit the design to those two fonts.



A restrained palette will outperform a clever one every time.

Contrast ratio

The measurable difference in lightness between foreground and background. WCAG AA requires 4.5:1 for body text — anything less fails accessibility.

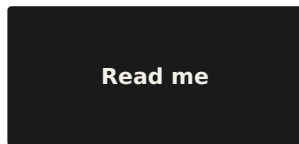
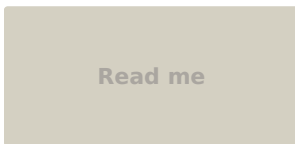
USE WHEN text feels hard to read, especially light gray on white or pale text on colored backgrounds.

— PROMPT

Ensure all text has at least 4.5:1 contrast against its background to meet WCAG AA accessibility standards.

2.1:1 (FAILS)

12:1 (PASSES)



Tonal range

How light to how dark your palette goes. A wider range gives you more drama and clearer hierarchy. A narrow range looks flat and muddy.

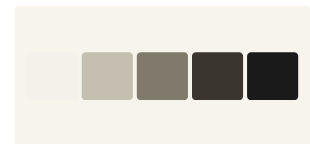
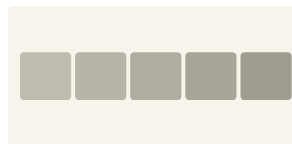
USE WHEN everything looks washed out or middle-gray, with no sense of depth or emphasis.

— PROMPT

Expand the tonal range of the palette. Include a near-white, a near-black, and three steps in between.

NARROW

WIDE



Accent color

The one color that pops. Used sparingly — on CTAs, key highlights, the occasional emphasis. Lose its scarcity and you lose its power.

USE WHEN you have five "important" colors fighting for attention and nothing actually feels important.

— PROMPT

Pick one accent color. Use it only for primary CTAs and key highlights. Everything else stays neutral.

FIVE ACCENTS

ONE ACCENT



Restrained palette

Two or three core colors max — usually two neutrals (light + dark) and one accent. Discipline beats variety in nine out of ten interfaces.

USE WHEN the design has 5+ colors and feels chaotic, like every section was designed by a different person.

— PROMPT

Reduce the palette to three colors total: one light neutral, one dark neutral, and one accent. Remove everything else.

SEVEN HUES

THREE



Six common requests, rewritten with the lexicon. Steal these directly.

- Translation Table

Vague → Specific

~~"Make it look better"~~

Establish stronger hierarchy. Use an 8-point spacing scale. Constrain measure to 65 characters.

~~"The buttons look weird"~~

Increase touch targets to 44px. Add 16px horizontal padding. Optically center the icon with the label.

~~"Make it feel premium"~~

Increase negative space around primary elements. Tighten tracking on headlines. Reduce density.

~~"Fix the spacing"~~

Establish vertical rhythm between sections. Snap padding to the 8-point scale.

~~"Make it work on mobile"~~

Mobile-first responsive. Breakpoints at 375 / 768 / 1024. Fluid typography between them.

~~"Too much going on"~~

Reduce density. Restrain the palette to two neutrals plus one accent. Constrain max-width.

If you only remember four things, remember these.

i.

Specify the scale. "8-point grid", "1.25 type ratio". Concrete numbers get enforced consistently.

ii.

Name the breakpoints. Don't say "make it work on mobile." Say "at 375px width."

iii.

Use "constrain." Magic word for things sprawling. Constrain max-width, measure, container.

iv.

Padding ≠ margin. Inside vs outside. They fix different problems. Be precise about which.

Starter *Template*

A copy-paste prompt scaffold. Fill in the orange blanks. Get a real designer brief in seconds.

- PROJECT

Build a **page type** for **audience**.

The goal is to **primary action**.

- COMPONENTS

Include: **component list**.

e.g., navbar, hero section, card grid, footer

- LAYOUT & SPACING

Use an 8-point spacing scale.

Constrain the container to max-width: **1200px**.

Establish clear hierarchy between H1 and subhead.

- TYPOGRAPHY

Pair a **serif/sans** headline with a **serif/sans** body.

Use only two font weights: Regular and Bold.

Constrain measure to ~65 characters.

- COLOR

Restrained palette: light neutral, dark neutral, one accent (**color**).

Ensure 4.5:1 contrast on all text.

- RESPONSIVE

Build mobile-first at 375 / 768 / 1024 breakpoints.

Touch targets minimum 44px on mobile.

- STATES

Design all button states: default, hover, disabled, loading.

Forms get focused, error, and success states.

How to use it

Paste this at the top of any Claude Code session. Replace the orange blanks with your specifics. Delete sections you don't need. The structure handles the rest.

Why it works

Real design briefs answer all of these questions before code gets written. Most prompts skip them. Filling in the blanks gives Claude the same context a designer would have.