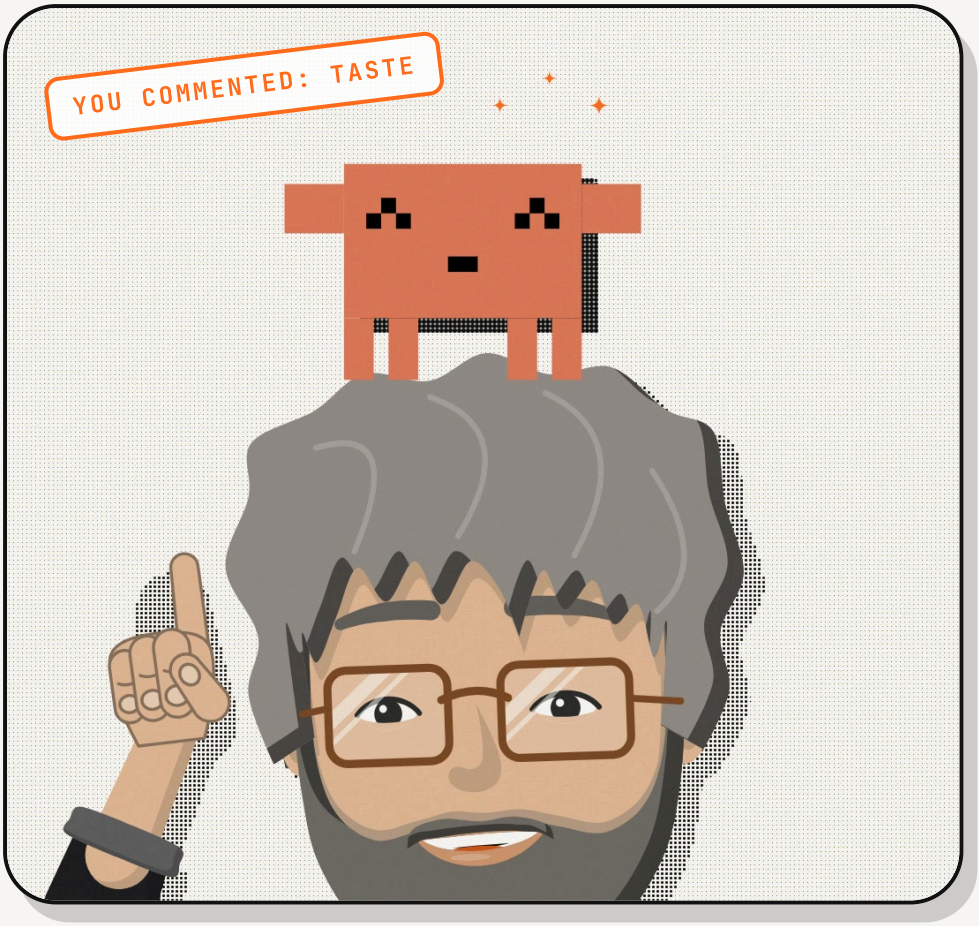
TASTE

FREE GUIDE



Design school for Claude: how to teach it *some taste*

The full course behind the reel: what causes the AI look, the four design skills in depth with ten use cases each, the best combinations as step-by-step workflows, and every prompt to copy. Written like a workbook: read, try, tick off.

// 01

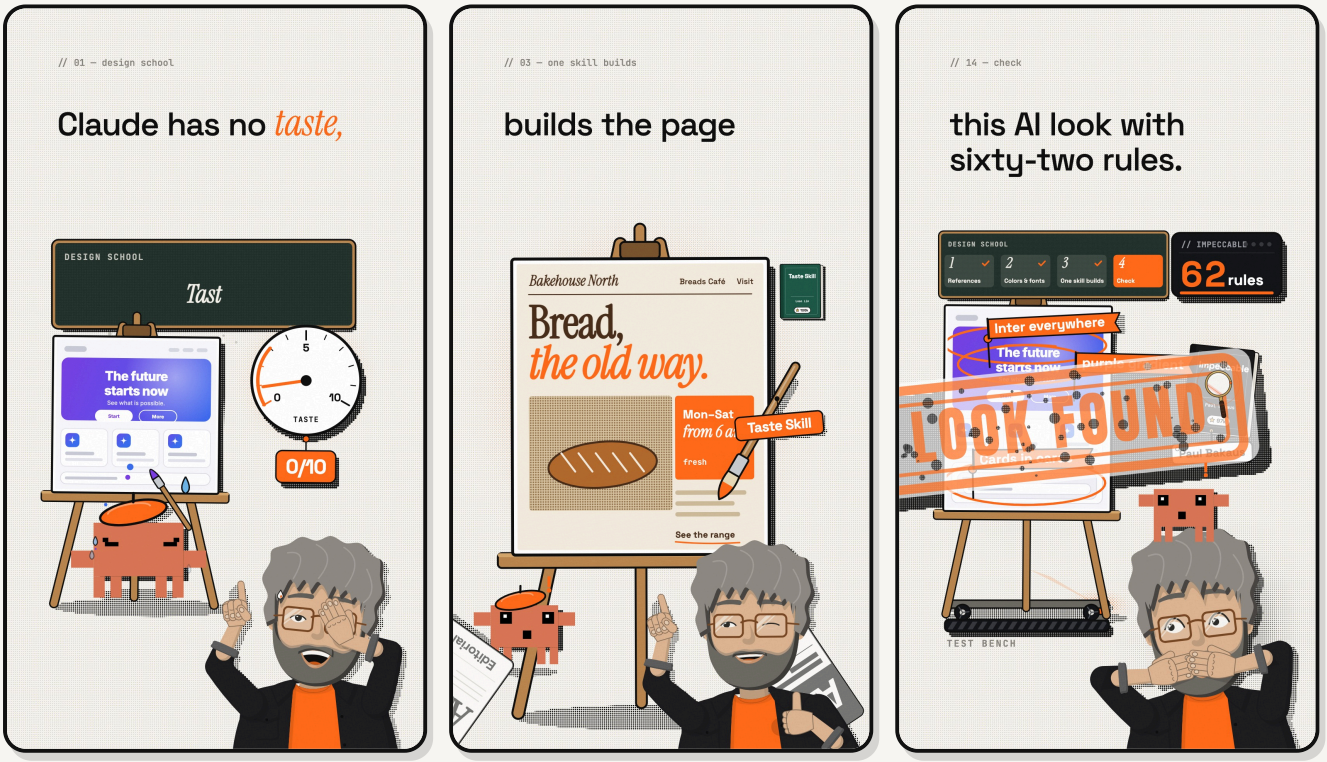
How to use this workbook

In this workbook I am your tutor. You don't need to know how to code. Every lesson follows the same pattern: **understand first, then try it, then remember it.**

LESSON	WHAT YOU LEARN	TIME
0 Search data	Why this topic is exploding right now	3 min
1 Basics	What a skill is and why Claude builds the AI look	10 min
2 Input	UI UX Pro Max: colors, fonts and rules for your industry	15 min
3 Build	Taste Skill: build a page with a clear direction	45 min
4 Motion	Skills by Emil Kowalski: animations that fit	10 min
5 Review	Impeccable: find and fix the AI look	15 min
6 Combine	Six ready-made workflows for typical jobs	20 min

Your learning goal: by the end you build a page with Claude that looks like your brand, not like AI, and you know which skill does what and when.

// FROM THE VIDEO



// 02

Lesson 0: What the search data reveals

Before we start, a look at demand. I analysed the topics with DataForSEO: Google search volume per month, average of the last twelve months, as of 31 October 2026.

SEARCH TERM	USA	GERMANY	TREND
claude code	550,000	135,000	almost tripled in a year
claude design	135,000	33,100	practically zero a year ago
ai slop	60,500	8,100	high and stable
claude skills	40,500	8,100	more than doubled
claude code skills	12,100	2,900	growing
ui ux pro max (skill)	5,400	1,300	brand new
taste skill	2,400	880	brand new
impeccable skill	1,900	–	brand new

What this means for you:

- **Designing with Claude is the topic of the moment.** "claude design" went from almost nothing to 135,000 searches a month in the US within twelve months.
- **People mostly ask how.** The most common questions: how do I add skills to Claude Code, how do I use them, which are the best, and "Impeccable vs Taste Skill". This workbook answers exactly that.
- **"AI slop" is an established term.** More than 60,000 searches a month in the US show that people notice the interchangeable AI look. Avoid it and you stand out.

// 03

Lesson 1: Basics

What is a skill?

A skill is a set of instructions in a folder, usually a `SKILL.md` file, often with scripts and reference files. At first Claude only reads the name and a short description. It loads the full content only when a task fits or when you call it with `/name`.

Good to know: according to the official Claude Code docs, the list of all skill descriptions has a fixed budget of **1 % of the context window**. Install a lot of skills and Claude Code drops descriptions, starting with the ones you use least. Then Claude finds the right skill less reliably. `/skill-doctor` shows you which skills you never use.

Why does Claude build the AI look?

Without guidance, a language model goes for the average of every page it has seen. The Impeccable README names the typical tells:

- Inter for everything
- Purple-to-blue gradients
- Cards nested in cards
- Gray text on colored backgrounds
- A rounded-square icon tile above every heading



THE MENTOR SAYS

Claude doesn't have bad taste. It just never got any. A skill gives it the decisions nobody else makes.

The model: four roles

Remember this picture, it carries the whole course: **input → build → motion → review.** Each role has a skill, and for each job only one sits at the wheel.

ROLE	SKILL	THE QUESTION IT ANSWERS
Input	UI UX Pro Max	Which colors, fonts and rules fit my industry?
Build	Taste Skill	What does a page with one clear idea look like?
Motion	Emil Kowalski	Which animation goes where, how fast, with which curve?
Review	Impeccable	Where is the AI look still hiding, what do I fix first?

Quick check: can you list the four roles from memory? Then move on.

// 04

Lesson 2: UI UX Pro Max, the input

What it is: a searchable design database for Claude, by Next Level Builder. According to the README: 79 UI styles, 192 color palettes, 74 font pairings, 25 chart types, 119 UX guidelines, 22 tech stacks and 192 industry rules. About 143,000 stars on GitHub.

How it works: Claude calls a small search script. For a new page it creates a complete design system. On request it saves it permanently as `design-system/<project>/MASTER.md`. Exceptions for single pages live in `design-system/<project>/pages/`.

Install:

```
/plugin marketplace add nextlevelbuilder/ui-ux-pro-max-skill
/plugin install ui-ux-pro-max@ui-ux-pro-max-skill
```

Top 10 use cases

#	SITUATION	HOW TO USE IT
1	New page for an industry	Let it create a design system for your industry. The 192 rules range from dental and legal to restaurants and SaaS.
2	Project with several pages	Save the design system as MASTER.md so every page follows the same rules.
3	One page should look different	Add a page override (for example the dashboard), MASTER stays untouched.
4	Find matching colors	Search the color palettes by mood, for example "warm artisanal".
5	Find matching fonts	Search font pairings by mood, with the Google Fonts import included.
6	Structure a landing page	Search the landing page patterns: hero, social proof, pricing.
7	Dashboard with numbers	Pick the right chart type and a denser spacing scale.
8	Accessibility	Query the UX rules: contrast 4.5:1, touch targets 44 × 44 px, visible focus.
9	Bolder or calmer	Set the dials for variance, motion and density from 1 to 10.
10	A specific stack	Get implementation rules for your stack, for example Next.js, Vue, Flutter or Tailwind.

Try it

```
Use UI UX Pro Max and create a design system for [industry, e.g. a bakery in
Brooklyn]. Audience: [who]. Save it permanently in the project and show me the colors
and fonts before you continue.
```

Remember: rules first, then build. Build without input and you get the average.

// 05

Lesson 3: Taste Skill, the build

What it is: a skill that gives Claude an art direction, by Leon Lin. Built from a study of 247 websites (Awwwards Sites of the Day and One Page Love). About 106,000 stars on GitHub.

How it works: Claude reads the brief. A script deals three of 20 researched directions and Claude picks one. It finds one idea in a sentence, builds with real pictures from open archives (credited), checks the page in a real browser at three widths, and fixes the top three problems in up to three rounds.

What you need to know: according to the README a run takes 30 to 45 minutes and a few hundred thousand tokens. You need Node 22 or newer and Chrome, Edge or Chromium.

Install:

```
npx skills add https://github.com/Leonxlnx/taste-skill --skill "design-taste-
frontend"
```

Top 10 use cases

#	SITUATION	HOW TO USE IT
1	New landing page	The standard run: three directions, one idea, build, check in the browser.
2	Portfolio, event, restaurant	Exactly these "pages that are read and scrolled" are the skill's core job.
3	Dashboard or admin area	A separate app run with its own rules: quiet panels, color spent on data.
4	Settings, billing, inbox	Also app screens, with a check made for apps.
5	Product site that shows your app	The page follows the page rules, the app fragments on it follow the app rules.
6	Fixed brand colors and fonts	With `--keep palette,type` it only changes the composition, your brand stays.
7	Redesign an existing page	Measure the "before", keep the stack and content, then compare the "after".
8	Just an honest review	Review mode stops after the critique and names the three changes that move the page most.
9	A single section	For components it matches the surrounding design and still checks the result.
10	No photos of your own	Pictures from Wikimedia Commons, Openverse or the Met, only public domain or CC, credited.

Try it

```
Use Taste Skill and build a landing page for [project]. Follow the design system in
the project and the references [Image #1] and [Image #2]. Use only real content:
[facts, prices, opening hours]. Don't invent numbers or testimonials.
```

Remember: one skill builds. Two building skills on the same page pull in different directions.

// 06

Lesson 4: Skills by Emil Kowalski, the motion

What it is: a collection of skills for animation and interface details, by Emil Kowalski. According to his README they are based on his years at Vercel and Linear. About 52,000 stars on GitHub.

Why it matters: according to the README, agents often pick the wrong curve, for example `ease-in` for something that enters. The right one there is `ease-out`: fast in, soft landing. These small things add up and decide whether a page feels premium.

Install:

```
npx skills@latest add emilkowalski/skills
```

Top 10 use cases

#	SITUATION	SKILL
1	A new animation from scratch	`animate` picks the curve, duration and properties.
2	Review existing animations strictly	`review-animations` checks against his rules.
3	Clean up a whole codebase	`improve-animations` writes a plan any agent can execute.
4	Where is motion worth it at all?	`find-animation-opportunities` finds spots and also says what to leave still.
5	Too much motion	`remove-animations` removes things like fade-ins on every section.
6	Janky animations	`animation-performance` finds where frames are dropped.
7	You can't describe what you want	`animation-vocabulary` gives you the right words for the prompt.
8	Web app should feel native on a phone	`mobile-native` fixes tap delays, the 100vh bug and zoom on input fields.
9	Test the interface with real data	`break-ui` tries long names, empty lists, huge numbers.
10	Compare several versions	`prototype` builds versions of a component with a switcher.

Also in the package: `apple-design` (Apple's design principles for the web), `pick-ui-library` (proven libraries instead of hand-rolled ones) and `animate-expo` (React Native).

Remember: things that enter get `ease-out`. And not everything has to move.

// 07

Lesson 5: Impeccable, the review

What it is: "The design vocabulary for your coding agent", by Paul Bakaus, who according to his GitHub profile created jQuery UI. Impeccable brings 24 commands and a slop detector with 62 deterministic rules that run without AI and without an API key. About 97,000 stars on GitHub.

Install:

```
npx impeccable install
```

Then reload your tool and run `/impeccable init` once. The installer offers a design hook that checks every file Claude edits. Accept it.

Top 10 use cases

#	SITUATION	COMMAND
1	Set up a project	`/impeccable init` writes PRODUCT.md, `/impeccable document` writes DESIGN.md.
2	Scored design critique	`/impeccable critique` scores and ranks what to fix first.
3	Technical audit	`/impeccable audit`: accessibility, performance, theming, responsive, ranked P0 to P3.
4	Final pass before shipping	`/impeccable polish`
5	Page is too safe or too loud	`/impeccable bolder` or `/impeccable quieter`
6	Page is cluttered	`/impeccable distill` removes what the page doesn't need.
7	Type, color or spacing is off	`typeset`, `colorize` or `layout`, aimed at one area
8	Edge cases and mobile	`harden` (errors, i18n, overflow) and `adapt` (screen sizes)
9	Variants right in the browser	`/impeccable live`: pick an element, cycle variants, accept one
10	Checks without an agent, also for teams	`npx impeccable detect src/`, with `--github` as GitHub Actions annotations

Tip from the README: target one component, not the whole page. `/impeccable critique the pricing cards` gives better results than a glance at everything.

Remember: Impeccable gets the last word before anything goes live.

// 08

Lesson 6: Combinations that work

Now we put the roles together. Each workflow is written so you can use it as is.

Workflow A: New landing page

1. Drag three screenshots of sites you like into Claude Code or paste them with Ctrl+V. They show up as `[Image #1]`, `[Image #2]`.
2. UI UX Pro Max creates the design system and saves it as MASTER.md.
3. Taste Skill builds only the top section of the page first.
4. Run `/impeccable critique` on that section and fix the first three points.
5. Taste Skill builds the rest, Emil's `review-animations` checks the motion.
6. Finish with `/impeccable audit` and `/impeccable polish`.

Workflow B: An existing page looks like AI

1. `npx impeccable detect https://your-site.com` shows which tells it finds.
2. `/impeccable critique` ranks the problems.
3. Fix them one by one: `typeset` for type, `colorize` for color, `distill` against clutter.
4. Use Taste Skill only if the whole page should be redone, in redesign mode with a before/after comparison.

Workflow C: Dashboard or app screens

1. UI UX Pro Max with high density and a chart recommendation.
2. Taste Skill builds the interface in its app run.
3. Emil's `break-ui` tests with extreme data, `mobile-native` takes care of the phone view.
4. `/impeccable audit` before going live.

Workflow D: Fixed brand colors

1. `/impeccable document` captures your existing design in DESIGN.md.
2. Taste Skill with `--keep palette,type`: your colors and fonts stay, only the composition is new.
3. Impeccable reviews. If Inter is your brand font, allow it: `npx impeccable ignores add-value overused-font Inter --reason "Brand font"`.

Workflow E: Agency with several clients

1. One project folder per client with its own MASTER.md (UI UX Pro Max) or DESIGN.md (Impeccable).
2. Always the same building skill, so quality stays comparable.
3. `npx impeccable detect --github` in the pipeline: every change is checked automatically.

Workflow F: Adding animations

1. `find-animation-opportunities` shows where motion helps.
2. `animate` builds it with the right curve.
3. `remove-animations` and `animation-performance` clean up.

Better avoid these combinations

COMBINATION	WHY NOT
Taste Skill + Impeccable building at the same time	A hands-on test on DEV Community found overlapping, partly conflicting rules. Build first, then review.
frontend-design + Impeccable at the same time	According to its own README, Impeccable started from frontend-design: the same idea twice.
OpenDesign + the same skills installed separately	OpenDesign ships copies of taste-skill, impeccable and frontend-design: duplicates in your skill list.
Impeccable `animate` + Emil's `animate`	Two animation rulebooks on the same spot. Pick one.

How to make sure only one builds: call the building skill by name ("Use Taste Skill to ..."). Skills you rarely need can be set to `"user-invocable-only"` with `skillOverrides` in the Claude Code settings. Then Claude only starts them when you type `/name`.

// 09

Guide: Your first page in 60 minutes

MINUTE	STEP	WHAT YOU DO
0-10	Install	Install the four skills, restart Claude Code.
10-15	References	Collect three screenshots, for example on Awwwards or Dribbble, and drag them into Claude Code.
15-20	Analysis	Let Claude describe what the references do well (prompt below).
20-25	Input	UI UX Pro Max: design system for your industry, saved as MASTER.md.
25-45	Build	Taste Skill: only the top section of the page.
45-55	Review	`/impeccable critique`, fix the first three points.
55-60	Polish	`review-animations`, then `/impeccable polish`.

// 10

Prompts to copy

Analyse references:

```
Look at [Image #1] and [Image #2]. Describe in 5 points what these pages do well (type, colors, spacing, images, structure). Don't build anything yet, I want to see your analysis first.
```

Design system for your industry:

```
Use UI UX Pro Max and create a design system for [industry]. Audience: [who]. Style: [calm / bold]. Save it permanently in the project and show me the colors and fonts before you continue.
```

Build only the top section:

```
Use Taste Skill and build only the top section of the home page for [project]. Follow the design system and [Image #1]. Show me the result in the browser before you build the rest.
```

Fixed brand colors:

```
Our brand only uses these colors: [hex values] and these fonts: [names]. Don't change them. Only redesign the layout, images and spacing.
```

Check animations:

```
Review all animations on this page with the skills by Emil Kowalski. List what is too slow, too much or uses the wrong curve, and suggest fixes.
```

Test with real data:

```
Use break-ui and test this page with extreme data: very long names, empty lists, very large numbers. Show me what breaks.
```

Honest critique:

```
/impeccable critique the home page
```

// 11

Common problems and fixes

PROBLEM	CAUSE	FIX
Claude doesn't use the skill	Description doesn't match or was dropped	Call the skill with `/name`, run `/skill-doctor`
Page still looks like AI	No references, no fixed colors	Go back to lesson 2 and step 1 of workflow A
Page invents features and numbers	No real facts in the brief	Give the facts and write "don't invent anything"
Usage limit runs out fast	Whole page built at once	Build section by section
Taste Skill doesn't check in the browser	No Node 22 or no Chrome	Update Node, install Chrome or Edge
Impeccable complains about your brand font	The overused-font rule	Allow the font with `ignores add-value`

// 12

Alternatives if one doesn’t fit

SKILL	ROLE	NOTE
Hallmark	builds	By Together AI. 50 themes, checks at eight widths. Instead of Taste Skill, not in addition.
frontend-design	builds	Official skill by Anthropic, one of the most installed design skills on skills.sh.
Awesome DESIGN.md	input	84 ready-made DESIGN.md files. As a learning template only, see the risks.
web-design-guidelines	reviews	By Vercel. 100+ rules for accessibility, performance and UX.

// 13

Limits and risks

- **Don't copy other people's designs.** Screenshots are inspiration. Rebuilding someone else's site or brand one to one can infringe trademark and copyright. You are responsible for the rights to other people's logos, images and text.
- **Check image rights.** Taste Skill uses images from open archives and puts the credits on the page. Keep those credits or replace the images with your own.
- **Install other people's code consciously.** Every skill is code or instructions written by someone else. Look at the repo before you install, especially hooks that run automatically.
- **Stars are not proof of quality.** There is no independent comparison. Stars show attention, not fitness for your project.
- **Invented content.** Building skills can make up features or numbers. Read everything before the page goes live.

// 14

Check your knowledge

Answer the questions before you read the solution.

1. **In which order do the four roles work?** Answer: input, build, motion, review.
2. **Why not two building skills at once?** Answer: their rules overlap and partly conflict, each pulls in its own direction.
3. **Which curve does an element get when it enters?** Answer: `ease-out`.
4. **Where does UI UX Pro Max save your design system?** Answer: in `design-system/<project>/MASTER.md`.
5. **What does `/skill-doctor` show you?** Answer: which skills you never use and what they cost in context.

- [ ] I know the four roles and their skills
- [ ] I have installed the four skills
- [ ] I have created a design system for my industry
- [ ] I have built one section with Taste Skill
- [ ] I have reviewed and improved it with Impeccable
- [ ] I have checked the animations with Emil's skills

// 15

My tip

Start small: one section, three screenshots, fixed colors. Once the top section looks good, the rest builds on it. And always give Impeccable the last word before anything goes live.

All links, commands and numbers checked on 31 October 2026 in the repos and on the official pages. Search volume: DataForSEO (Google Ads data), monthly average of the last twelve months.



THE MENTOR SAYS

Give Claude good examples, fixed rules and an honest reviewer. Then it has taste.

— @reposprites

Want to make animated videos like this too? Visit my website and get the reel engine.

reposprites.com ↗