

Claude Code HTML Guide

Stop Using Markdown. Start Creating Interactive HTML.

Setup Guide · 20 Live Examples · 10 Ready-to-Use Prompts

Why HTML Instead of Markdown?

Every markdown file Claude creates — nobody reads it. Not your team. Not your boss. Not even you after day one. HTML changes that completely. Same Claude. Same prompt. Completely different output.

✗ Markdown

- Plain text walls nobody reads
- Hard to share with your team
- No charts, colors or diagrams
- ASCII art instead of real visuals
- No interaction or live controls

✓ HTML

- Interactive dashboards people open
- Share with one link instantly
- Charts, tables, color, SVG diagrams
- Real vector illustrations
- Sliders, knobs, export buttons

Setting Up Claude Code in Your Terminal

 **Time:** Approximately 10 minutes from zero to your first HTML output.

Step 1 — Install Node.js

Claude Code requires Node.js version 18 or higher.

Mac (using Homebrew):

```
brew install node
```

Windows — download installer from nodejs.org and run it.

Linux (Ubuntu/Debian):

```
sudo apt update && sudo apt install nodejs npm
```

Verify:

```
node --version
```

Step 2 — Install Claude Code

```
npm install -g @anthropic-ai/claude-code
```

Verify:

```
claude --version
```

Step 3 — Authenticate

Run Claude Code for the first time. It will prompt you to log in via browser.

```
claude
```

You need a Claude Pro or Max plan for full Claude Code access.

Step 4 — Navigate to Your Project

```
cd /your/project/folder
```

```
claude
```

Step 5 — Generate Your First HTML File

 **Try this first:** Create an HTML file summarizing what this project does. Include a visual overview, key components, and a quick start section. Make it easy to read and share.

Open the output in your browser:

```
open output.html      # Mac
start output.html     # Windows
xdg-open output.html  # Linux
```

20 Live Examples — Open Any in Your Browser

These are real HTML files generated by Claude Code. Click any link to open the live example. Each one shows what a real output looks like across different use cases.

Full gallery: thariqs.github.io/html-effectiveness

01 — Exploration & Planning

When you're not sure what direction to take. Ask Claude to fan out across several options and lay them next to each other so you can point at one.

Live Example	What It Shows
Three Code Approaches	Side by side comparison of three ways to solve the same problem with trade-offs called out inline.
Visual Design Directions	Layout and palette options rendered live so you can react to them not imagine them.
Implementation Plan	Milestones on a timeline, data flow diagram, inline mockups, risky code, and a risk table.

02 — Code Review & Understanding

Diffs and call graphs are spatial information. Markdown flattens them. Let Claude render the change as an annotated diff.

Live Example	What It Shows
Annotated Pull Request	A diff rendered with margin notes, severity tags and jump links. Easier to scan than scrolling a terminal.
PR Writeup for Reviewers	Motivation, before and after, a file by file tour with the why, and where to focus the review.
Module Map	An unfamiliar package drawn as boxes and arrows with the hot path highlighted and entry points listed.

03 — Design

HTML is the natural format for talking about design systems. Tokens become swatches. Components become contact sheets.

Live Example	What It Shows
Living Design System	Colors, type scale and spacing tokens pulled from a repo and rendered as swatches you can copy from.
Component Variants	Every size, state and intent of one component laid out on a single sheet for review.

04 — Prototyping

Motion and interaction can't be described. Only felt. A throwaway page with the real easing curve tells you in five seconds what a paragraph of prose never could.

Live Example	What It Shows
Animation Sandbox	The transition in isolation with sliders for duration and easing so you can tune it before wiring it in.
Clickable Flow	Four screens linked together. Enough fidelity to feel whether the interaction is right.

05 — Illustrations & Diagrams

Inline SVG gives Claude a real pen. Ask for figures for a post or a flowchart of a process and get vector art you can tweak by hand.

Live Example	What It Shows
SVG Figure Sheet	Diagrams for a blog post drawn inline so they can be tweaked and copied out one by one.
Annotated Flowchart	A deploy pipeline drawn as a real flowchart. Click any step to see what runs, timings, and failure paths.

06 — Decks & Presentations

A handful of section tags and twenty lines of JS is a slide deck. Point Claude at a Slack thread and get something you can arrow-key through in a meeting.

Live Example	What It Shows
Arrow-key Slide Deck	A short presentation as one HTML file. Left and right to navigate. No build step. No Keynote.

07 — Research & Learning

An explainer with collapsible sections, tabbed code samples, and a glossary reads very differently from the same words dumped linearly.

Live Example	What It Shows
How a Feature Works	TL;DR box, collapsible request path steps, tabbed config snippets, and an FAQ for understanding rate limiting.
Concept Explainer	Consistent hashing taught with a live ring you can add and remove nodes from, a comparison table, and a glossary.

08 — Reports

Recurring documents — status updates, post-mortems — benefit most from structure and color. A small chart turns something people skim into something they actually read.

Live Example	What It Shows
Weekly Status Report	What shipped, what slipped, and a small chart. Formatted for a quick skim on Monday morning.
Incident Timeline	A post-mortem with a minute by minute timeline, log excerpts, and the follow up checklist.

09 — Custom Editing Interfaces

Sometimes it's hard to describe what you want in a text box. Ask for a throwaway editor — and always end with an export button that turns what you did back into something you can paste into Claude Code.

Live Example	What It Shows
Ticket Triage Board	Drag thirty tickets across Now / Next / Later / Cut then copy the final ordering out as markdown.
Feature Flag Editor	Toggles grouped by area, dependency warnings when a prerequisite is off, and a copy diff button.
Prompt Tuner	Editable template on the left with variable slots highlighted. Three sample inputs re-render live as you type.

10 Ready-to-Use Prompts

Copy and paste these directly into Claude Code. Each one is designed to generate a stunning HTML output immediately.

Use Case	Prompt to Use
Project Plan	Create a thorough implementation plan in an HTML file. Include mockups, data flow diagrams, and important code snippets. Add tabs or sections to make it easy to navigate.
Code Review	Help me review this PR by creating an HTML artifact. Render the diff with inline margin annotations, color code findings by severity, and add a summary at the top.
Feature Exploration	I'm not sure what direction to take this feature. Generate 4 distinctly different approaches in a single HTML file in a grid so I can compare them side by side.
Design Prototype	Create an HTML file with interactive sliders and color pickers to prototype this animation. Add a copy button that exports the final parameters.
Technical Explainer	I don't understand how our rate limiter works. Read the relevant code and produce a single HTML page with a flow diagram, 3-4 annotated code snippets, and a gotchas section.
Slide Deck	Create a short presentation as a single HTML file I can arrow-key through in a meeting. Pull from this document. No build step needed.
Weekly Report	Create a weekly status report as an HTML file. Summarize what shipped, what slipped, and what is blocked. Make it ready to share with leadership.
Ticket Triage	Make an HTML file with these tickets as draggable cards across Now / Next / Later / Cut columns. Pre-sort by priority. Add a copy as markdown button.
Prompt Tuner	Make a side-by-side HTML editor. Editable prompt on the left with variable slots highlighted. Three sample inputs on the right that re-render live. Add a copy button.
General Output	Make this as an HTML file. Organize it with tabs and sections. Add SVG diagrams where helpful. Make it mobile responsive and easy to share with a link.

Quick Tips for Better Results

- Always say HTML file or HTML artifact — Claude defaults to markdown otherwise.
- Ask for tabs and sections to make long documents easier to navigate.
- Ask for SVG diagrams instead of ASCII art for much cleaner visuals.
- Ask for mobile responsive if you plan to share the link with others.
- Always add a copy button to editing interfaces to export changes back into Claude Code.
- Upload to S3 or any static host to share with one link. No attachments needed.
- HTML takes 2-4x longer to generate than markdown. The results are worth it.
- Use the Frontend Design plugin in Claude Code for better styled outputs.



Note: The features that make HTML feel self-maintaining rely on Claude following multi-step instructions precisely. The more specific your prompt the better the output.

You were never limited by Claude's intelligence.

You were limited by the format you were asking for.

HTML just removed that limit.



Follow for more: More Claude Code tips, AI tools, and guides like this every week. Comment HTML on the reel if this guide helped you.