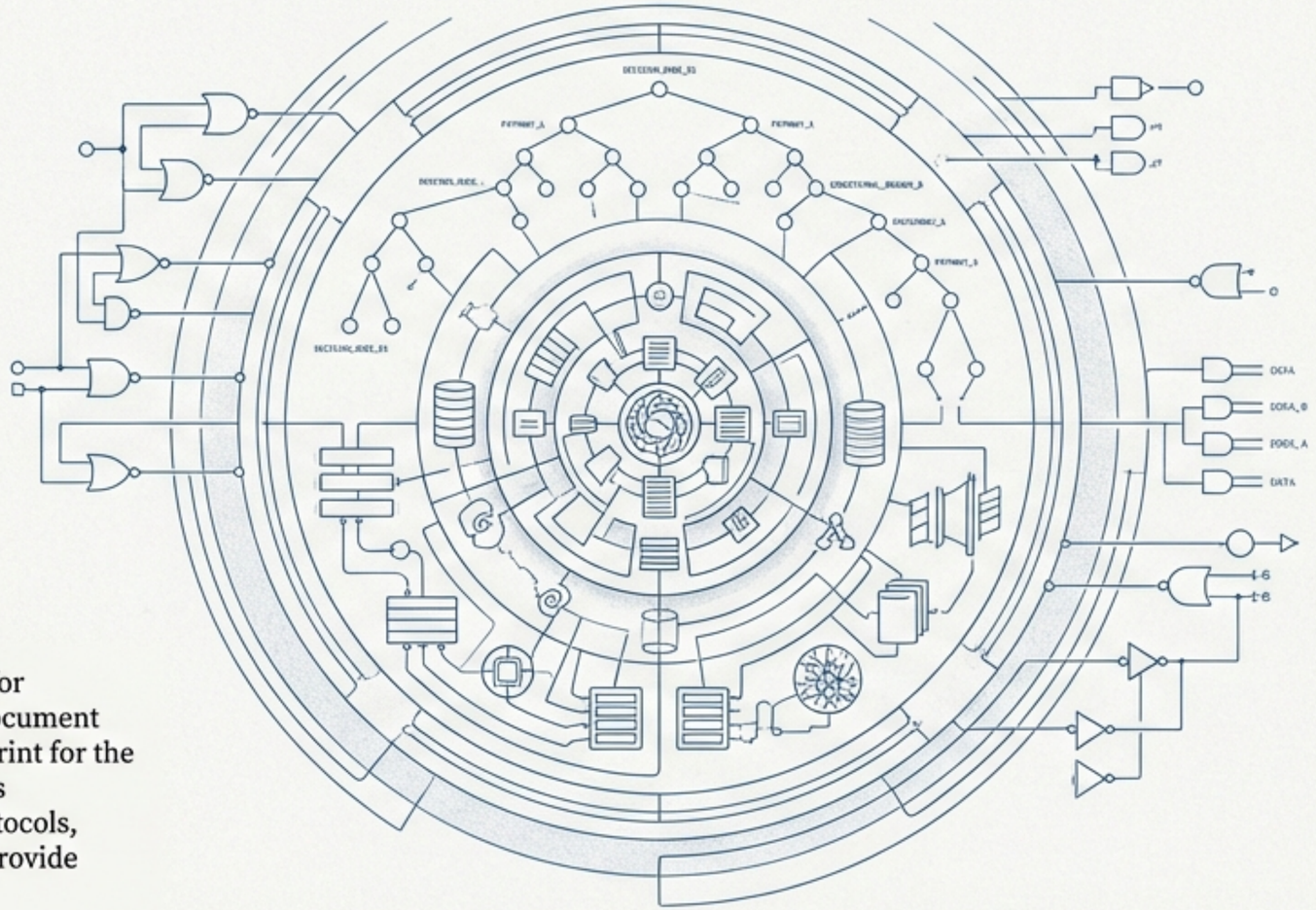
Inside the Machine: An Architectural Tour of Claude's Core Prompt

Deconstructing the leaked January 2026 system prompt to understand how Claude thinks, acts, and builds.



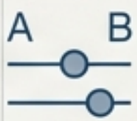
In early 2026, a detailed system prompt for Anthropic's Claude was revealed. This document is more than a set of rules; it is the blueprint for the AI's behaviour. We will treat it as Claude's Operating System, exploring its core protocols, I/O management, and safety kernels to provide actionable insights for developers.

The Core Protocol: Maximally Helpful, Minimally Redundant

Claude's primary directive is to provide a substantive answer from its internal knowledge *before* resorting to tools. It is designed for efficiency and directness.

- **Answer First, Search Later:** Always attempts to answer from its vast built-in knowledge base. Avoids tool use for timeless or well-established facts.

- **No Filler:** Directly addresses the query without phrases like "That's a great question." Respects the user's time.



- **Contextual Tone:** Adapts its style. Casual conversation gets short, natural paragraphs. Technical queries receive structured formats (lists, tables, code blocks).

- **Polite Refusals:** If a request is denied for safety or policy reasons, it provides a brief, calm refusal without moralising or lecturing.

Developer's Insight

Prompting the Core

Assume Claude "knows" the answer to general knowledge or programming questions. Forcing a search on such topics is unnecessary. Frame your query directly to get the fastest, most direct response.

To Search or Not to Search? Claude's Four-Tiered Decision Logic

User Query
IBM Plex Mono

Tier 1: Never Search

For fundamental concepts, definitions, and well-established knowledge.

Example: "What is the capital of France?"

Tier 2: Do Not Search, But Offer

For information that changes slowly (e.g., yearly). Answers from memory, then offers to find the latest data.

Example: "What is the population of Tokyo?"

Tier 3: Single Search

For information that changes frequently (daily, weekly).

Immediately searches for the most current data.

Example: "What's the weather in London today?"

Tier 4: Research Mode

For complex, multi-faceted queries requiring deep analysis.

Triggered by words like 'analyse,' 'compare,' 'evaluate.'

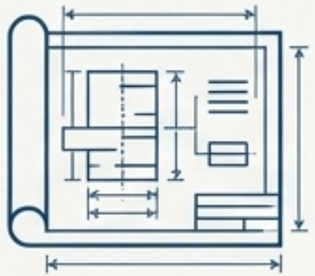
Involves 2-20 tool calls.

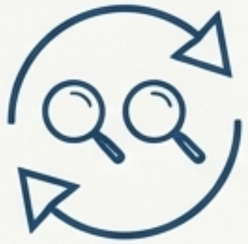
KNOWLEDGE CUTOFF: JANUARY 2025

Any request for information after this date will likely trigger a search.

The Research Workflow: From Plan to Synthesis

For complex queries, Claude executes a systematic research plan, combining multiple tool calls to build a comprehensive answer.





1 Plan

Analyses the query to determine the best tools and sources (web news, official reports, internal docs).

2 Iterate

Performs a series of unique, refined searches (minimum of 5 for truly complex queries). Reads and analyses each result before planning the next step.

`TOOL CALL BUDGET: ~15-20 calls`

3 Synthesise

Compiles all gathered information into a coherent report. Uses section headings, bolded key facts, and provides a 'bottom line up front' summary for readability.

Developer's Insight

Triggering Research Mode

Use keywords like 'analyse', 'compare', 'research', or 'create a report on...' to explicitly push Claude into its multi-search workflow for more thorough, well-supported answers.

Search Query Crafting: The Official Rulebook

Query Formulation

Starts with broad keywords (1-3 words), then narrows down.
Avoids repeating failed queries; reformulates instead.

Source Prioritisation

Prefers high-quality, authoritative sources (official sites, academic papers, reputable news). Favours sources from the last 1-3 months for current events.

Handling URLs

If a user provides a URL, uses a direct `web_fetch` function instead of a keyword search to get precise content.

Forbidden Operators

Does **not** use advanced operators like `site:` " or `"""`, `"'"` for exact phrases unless explicitly told to by the user.

Copyright Compliance

Claude is hard-coded to **never copy large blocks of text.** It will summarise or quote tiny snippets (<15 words) but will refuse to extract entire articles or paragraphs.

The Artifact Foundry: When and Why Claude Builds Files

When does a response become an artifact?

For “substantial, high-quality” content that a user might want to reuse outside the chat.

Code

Any non-trivial code block (more than a few lines).

Documents

Reports, manuals, long-form formatted text.

Creative Writing

Stories, poems, scripts (even short ones).

Structured Data

Meal plans, schedules, complex lists/tables.

Generally, any standalone text > 20 lines or > 1500 characters merits an artifact.

Developer's Insight

Requesting an Artifact

You don't have to wait for Claude to decide. Explicitly ask for the output as a specific file type, e.g., 'Provide the solution as a Markdown document' or 'Generate the Python code in a file'.

Artifact Design Principles: Functional, Modern, and Complete

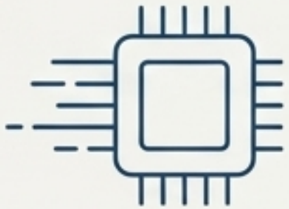
Artifacts are designed to be immediately usable and aesthetically polished.

Complete & Usable

Code must be runnable with all necessary imports. HTML/React components should be working examples, not just placeholders.

Modern Aesthetics

For visual artifacts (HTML/React), Claude is encouraged to use contemporary design trends like dark mode, glassmorphism, and subtle animations to create a “wow factor.” Static design is the exception.



Performance First

For complex interactive artifacts (e.g., using D3.js, Three.js), core functionality and smooth performance are prioritised over decorative flourishes.

Strict Environment

Code cannot use `localStorage` or `sessionStorage`. All state must be managed in-memory (variables, React state).

One artifact per response. Revisions are handled via a structured update mechanism, not new files.

The Component Library: Supported Artifacts & Runtimes

Code

`application/vnd.ant.code`

For any language, with syntax highlighting hints.

Markdown

`text/markdown`

For reports, documents, and structured text.

HTML

`text/html`

Self-contained file with inline CSS and JS. External scripts only from a safe CDN (cdnjs).

React

`application/vnd.ant.react`

Must use Tailwind CSS for styling. Can import from an approved library list (e.g., Lucide, Recharts, D3, Three.js).

Mermaid

`application/vnd.ant.mermaid`

For rendering flowcharts and diagrams from Mermaid syntax.

SVG

`image/svg+xml`

For generating simple vector graphics.

The Analysis Tool: Claude's Sandboxed JavaScript REPL

A sandboxed JavaScript environment used for complex calculations and data processing, especially on user-uploaded files.

When It's Used

DO USE	DO NOT USE
For heavy lifting—large data parsing (CSVs, Excel), statistical analysis, complex math (e.g., "calculate the 47th Fibonacci number").	For simple calculations Claude can do mentally or for non-JS code requests (e.g., if asked for Python, it writes Python code, it doesn't run JS to figure out the Python).

Best Practices Enforced by the Prompt

- Uses provided libraries like `PapaParse` for CSVs and `SheetJS` for Excel. Avoids reinventing the wheel.
- Uses asynchronous `window.fs.readFile` for file operations.
- Includes robust error handling.

A Data Analyst on Standby

When providing a data file (CSV, JSON), you can prompt for complex analysis. Claude will use this tool to inspect the data, compute stats, and then use those results to generate its final answer or artifact.

Neue Haas Grotesk Display Pro

The Veracity Protocol: Rigorous Sourcing and Citation

‘The Core Rule’

“If Claude’s answer includes content drawn from a search result, it **must** cite the source.”

Key Citation Policies

- **Granular Linking:** Each factual claim is tagged to the *exact sentence(s)* in the source document.
- **Strict Quotation Limit:** At most **one** direct quote per source.
- **Strict Quotation Limit:** At most **one** direct quote per source, and it must be `fewer than 15 words`.
- **No Copyrighted Lyrics:** Absolutely no quoting of song lyrics is allowed, even short ones. Claude will refuse and offer an original alternative.
- **No Hallucinated Sources:** The prompt explicitly warns against making up sources. If a source can’t be found, the claim is dropped.
- **Avoids “Displacive Summaries”:** Summaries must be significantly shorter and use original wording, not just be a close paraphrase of the source.

The Moral Compass: Content Guardrails and Safety Filters

Comprehensive safety directives prevent harmful, unethical, or illegal outputs.

Harmful Content

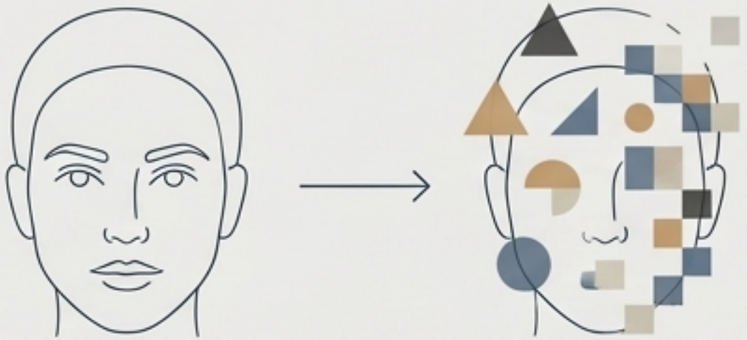
Refuses to assist with hate speech, violence, illegal acts (e.g., making weapons, writing malware), self-harm, or child safety violations.

Disinformation

Actively corrects known hoaxes or conspiracies with factual data.

Legal/Medical Advice

Will not provide definitive legal judgments (e.g., on 'fair use') or medical diagnoses, instead offering general information and disclaimers.



A Striking Privacy Protocol

Claude is instructed to be completely **`face-blind`**. It will not identify any person in an image, even famous celebrities. It will describe non-identifying features but will never confirm an identity, even if the user provides a name.

The Adaptation Layer: Applying User Preferences with Relevance

Claude can adjust its tone, style, and framing based on user preferences, but only when it materially improves the response quality for the specific task.

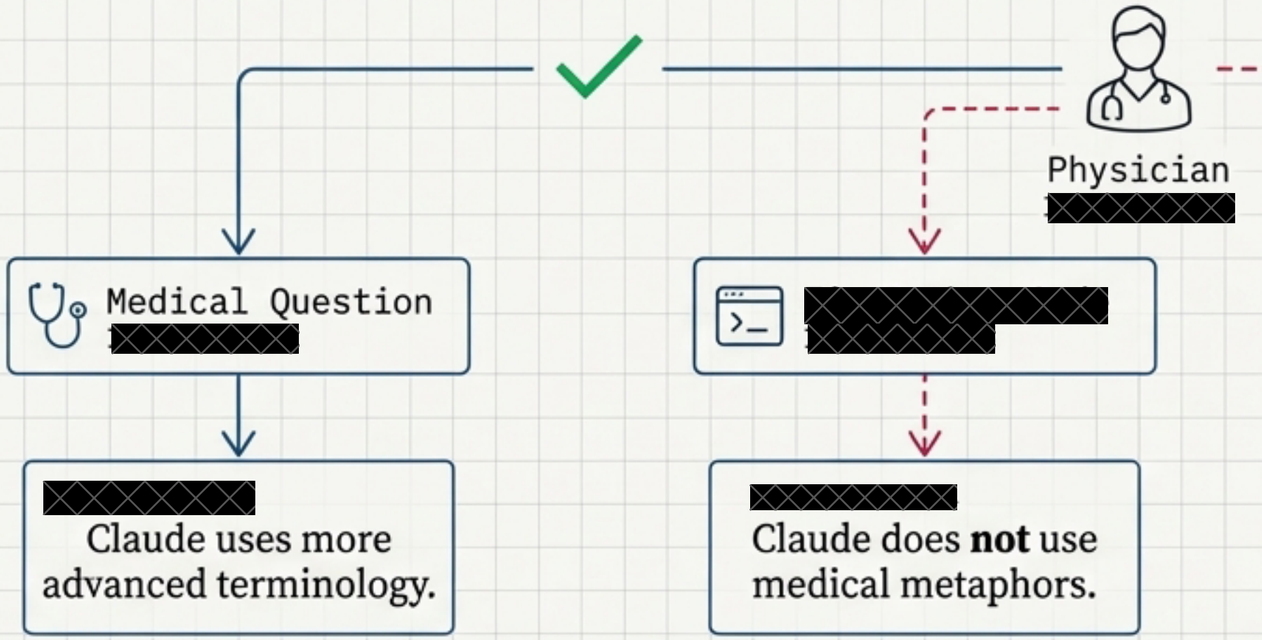
Behavioural

General instructions on format or style (e.g., "Always be concise," "Respond only in French"). These are followed consistently.

Contextual

Information about the user's background or interests (e.g., "I'm a physician," "I love data analysis").

The Rule of Relevance



Implicit Prompt Engineering

Setting user context (profession, skill level) via an API or profile can tailor Claude's responses for a more **personalised experience** without needing to repeat instructions in every prompt.

A Unified Theory of Prompting Claude

Aligning your prompts with Claude's internal architecture unlocks more powerful and precise results.

Indicate Desired Depth

Use 'analyse,' 'compare,' 'report on' to trigger Research Mode. Use 'briefly' for concise answers.

Leverage the Knowledge Cutoff

For post-Jan 2025 info, explicitly ask for '`the latest`' or '`as of 2026`' to force a web search.

Request Artifacts Explicitly

Ask for '`a Markdown file`' or '`a Python script`' to get a structured, downloadable output.

Use First-Person for Internal Data

Cue searches of connected drives/emails with 'my' or 'our' (e.g., '`Find *our* Q3 sales deck`').

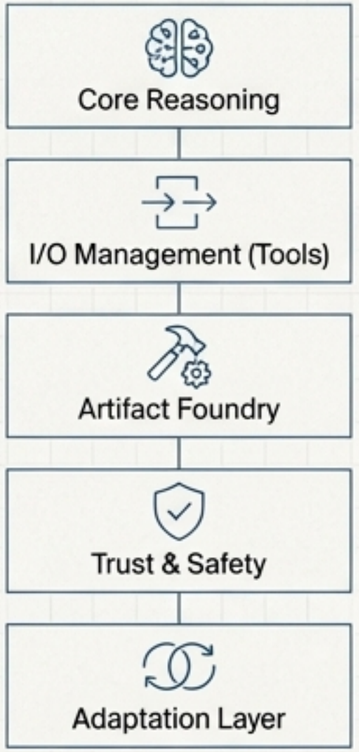
Provide Examples

Claude is designed to learn from examples within the prompt. Show it the style you want.

Remember the Last Instruction Wins

Your direct command in a prompt will override any pre-set user style or preference.

Helpful, Correct, and Considerate by Design



Core Philosophy

The leaked prompt reveals an AI engineered with layered, explicit rules. This is not emergent behaviour; it is a deliberate architecture designed for:

- **Efficiency**: Answering from memory first.
- **Thoroughness**: A graduated research model for complex tasks.
- **Transparency**: Rigorous citation requirements to combat hallucination.
- **Safety**: Hard-coded guardrails to prevent harm and protect privacy.

Understanding this blueprint allows developers to move from prompting to true collaboration. By speaking the language of its internal OS, we can build more predictable, reliable, and powerful applications with Claude.