

Every Claude Concept

75 / 75



Every Claude Concept

01 The engine

01 Artificial intelligence

Umbrella term for software that does things we used to think needed a human brain.

02 Machine learning

AI that learns patterns from millions of examples instead of hand-written rules.

03 Neural network

Layers of tiny math units passing numbers; learning adjusts the strength of each connection.

04 Large language model

A neural network trained on huge amounts of text to read and write language. Claude is one.

05 Pretraining

First training stage: the model guesses the next piece of text, trillions of times.

06 Parameters

The connection strengths (weights). A model's knowledge is spread across billions of them.

07 Tokens

Chunks of text a model reads and writes, about three quarters of a word each.

08 Next-token prediction

Claude writes one token at a time: score every option, pick one, repeat.

09 Context window

Everything Claude can see at once. Its working memory, big but finite.

10 Knowledge cutoff

The date after which a model hasn't seen any data.

11 Hallucination

A fluent, confident answer that's wrong. Check numbers, quotes and citations.

12 Temperature and sampling

How the next token is picked. Low temperature is focused, high is more varied.

13 Post-training

Second training stage that turns an autocomplete into an assistant that follows instructions.

14 RLHF

Reinforcement learning from human feedback: people rate answers, the model moves toward the preferred ones.

15 Constitutional AI

Anthropic's method: Claude learns to critique and improve its answers against a written set of principles.

16 Anthropic

The AI safety and research company that makes Claude.

17 Model family

Haiku (fast), Sonnet (balanced), Opus (harder problems), Fable (most demanding). Bigger costs more.

18 Multimodal

Claude can take in images, screenshots, charts, handwriting and PDFs, not just text.

02 Talking to Claude

19 Prompt

What you send Claude. Better input, better output.

20 System prompt

Hidden instructions from whoever built the app: who Claude is, its tools, its rules.

21 Prompt engineering

Writing prompts that work: give context, the audience, an example of good, and what to avoid.

22 Few-shot examples

Show a couple of finished examples and Claude matches their format and tone.

23 Extended thinking

Claude reasons step by step before answering, which helps with math, code and planning.

24 Effort

How hard Claude works on a task. Low is quick and cheap, high digs deeper.

25 Conversations are stateless

Claude re-reads the whole conversation every turn; it doesn't remember between messages.

26 Projects

A home base of files and instructions that every chat inside it can see.

27 Memory

Preferences and context Claude carries across conversations. You can view, edit or turn it off.

28 Styles

Change how Claude writes (concise, formal, your own voice) without changing what it knows.

29 Web search

Claude looks things up live and answers with citations, getting around the knowledge cutoff.

30 Research

Web search turned up: many searches over several minutes, cross-checked into a cited report.

03 Claude makes things

31 Artifacts

Apps, dashboards, diagrams and documents Claude builds live in a panel next to the chat.

32 Publishing and sharing

Artifacts are private by default; publish one to share a link, or keep it in your team.

33 Artifact capabilities

Artifacts that read live data, save shared state, know the viewer, or ask Claude questions.

34 Files and documents

Real spreadsheets, slide decks, Word documents and PDFs you can open in Office.

35 Claude Design

A canvas for screens, prototypes and decks built with your design system.

Every Claude Concept

04 Claude reaches out

- 36 **Tool use**
How Claude does things: it requests a tool, the app runs it, the result comes back.
- 37 **Agents and the agent loop**
Tool use in a loop: decide, act, check the result, repeat until the job is done.
- 38 **MCP**
Model Context Protocol: an open standard for connecting AI to tools and data. USB-C for AI.
- 39 **Connectors**
One-click MCP: connect email, calendar, docs and more so Claude can read and act in them.
- 40 **Skills**
Folders of instructions (and scripts) that teach Claude one job; loaded only when needed.
- 41 **Plugins**
One install that bundles skills, commands, agents, hooks and connectors.
- 42 **Claude in Chrome**
An extension that lets Claude navigate, click and fill forms in your own browser.
- 43 **Computer use**
Claude controls the mouse and keyboard from screenshots, so it can use apps with no API.
- 44 **Claude in Slack**
Tag Claude in a thread and it reads the context and helps without anyone switching apps.

05 Claude Code

- 45 **Claude Code**
Claude as an agent on your computer: reads files, edits code, runs commands and tests.
- 46 **Where it runs**
Terminal, the desktop app's Code tab, the web, VS Code and JetBrains. Same agent.

- 47 **CLAUDE.md**
A project file Claude reads every session: how to run, test, your conventions, what not to touch.
- 48 **Permission modes**
How much Claude can do without asking, from asking every time to bypass in a sandbox.
- 49 **Plan mode**
Read-only: Claude explores and writes a plan, and changes nothing until you approve.
- 50 **Slash commands**
Shortcuts typed with /. Some built in, and you can write your own.
- 51 **Hooks**
Your scripts that run at set moments, every single time, e.g. format code or block commands.
- 52 **Subagents**
Helpers with their own fresh context window that do a focused job and report back.
- 53 **Sessions and worktrees**
Run several sessions at once; worktrees give each its own copy of the repo on its own branch.
- 54 **Background and cloud sessions**
Sessions that keep working in the cloud while your laptop is closed; check in from your phone.
- 55 **Routines and loops**
Routines run tasks on a schedule; the loop command reruns a prompt on an interval.
- 56 **Compaction**
When the context window fills, Claude summarizes older history and keeps going.
- 57 **Output styles**
How Claude Code narrates its work: concise, explanatory, or a learning mode.
- 58 **Mods**
Small plugins that change the tool itself: status lines, panels, banners, notifications.

- 59 **Workflows**
Scripts that coordinate many agents at once for big, thorough jobs.
- 60 **Code review**
Claude checks a diff for bugs at the depth you choose; ultra runs a deep multi-agent review.
- 61 **Fast mode**
The same top model with faster output, for when waiting is the bottleneck.

06 Building with Claude

- 62 **The API**
How developers send messages to Claude from their own code (the Messages API).
- 63 **Pricing per token**
You pay per token in and out; output and bigger models cost more.
- 64 **Prompt caching**
Reuses the processed version of a repeated prompt, so requests are faster and much cheaper.
- 65 **Batch processing**
Send a big pile of requests at once and get them back later at a lower price.
- 66 **Agent SDK**
The engine behind Claude Code, for developers building their own agents.
- 67 **Managed Agents**
Anthropic hosts the agent and its sandbox, so you run agents without managing servers.
- 68 **Evals**
Test cases that measure how well your prompt or agent performs, so you know a change helped.

07 Keeping it safe

- 69 **Alignment**
Making sure AI does what people intend and stays honest and safe as it gets more capable.

- 70 **Prompt injection**
Hidden instructions in content Claude reads, like a web page or email, trying to hijack it.
- 71 **Jailbreaks**
Deliberate attempts to trick a model into ignoring its rules.
- 72 **Interpretability**
Looking inside a model to understand how it thinks. Neuroscience for AI.
- 73 **Responsible Scaling Policy**
Anthropic's commitment that stronger safety measures must exist before more capable models.
- 74 **Model welfare**
Research into whether models like Claude could have experiences that matter.
- 75 **Usage policy**
The rules for what Claude can and can't be used for.