

START [HERE](#)

Introduction to AI agents.

From simple chat to tools, workflows and supervised action.

[~/GUIDES/003](#)

Give the system a goal. Let it use tools. Stay in control.



00 / INTRODUCTION

From answers to actions.

A chatbot responds to your questions. An agent can take several steps and use tools to work towards a goal.

The word “agent” is used loosely, but the practical idea is straightforward: a system gets a task, chooses or follows actions, observes what happened and continues until it reaches an outcome or needs help. Its permissions and supervision matter just as much as its model.

THE SHORT VERSION

Model + instructions + tools + execution loop + boundaries. You still decide what the agent may access, change or publish.

IN THIS GUIDE

- 01 **What an agent needs**
Models, instructions, tools and state.
- 02 **How the loop works**
Observe, act, review and stop.
- 03 **Real use cases**
Where an agent saves actual effort.
- 04 **Giving a useful brief**
Goals, constraints and definition of done.
- 05 **Staying in control**
Permissions, approvals and checkpoints.
- 06 **Starting small**
A supervised workflow you can try.

01 / THE ANATOMY

Four pieces of an agent.

An agent is more than a model answering chat messages. Its surrounding system determines what it can really do.

01 / MODEL

The reasoning engine

Interprets the task, proposes steps and decides which supported action to request. Models can misunderstand or make mistakes.

02 / INSTRUCTIONS

The working brief

Defines the goal, scope, priorities and what counts as complete. Clear limits matter more than dramatic prompts.

03 / TOOLS

The capabilities

Search, read files, run code, use a browser or call an approved API. A tool must actually be connected and authorised.

04 / STATE

The working context

Conversation history, retrieved files, checkpoints and tool results help the system continue without losing its place.

SURROUNDING ALL FOUR

Guardrails, access controls, logs and human approvals keep an agent useful without giving it unlimited power.

Chat, workflow or agent?

Not every AI-assisted process needs autonomy. Choose the simplest arrangement that handles the job.

CHAT ASSISTANT

You drive each step

You ask for a draft, review it and manually make the next request. Useful for ideas, explanations and ordinary writing.

FIXED WORKFLOW

The sequence is predefined

A form is submitted, a document is summarised, then a draft is saved. The steps were mostly decided by the programmer.

AI AGENT

The system selects actions

You specify a goal; the agent can inspect results, choose relevant tools, adjust its plan and decide when to stop or request approval.

KEEP IT SIMPLE

If a standard script or single AI response solves the task, an agent may add cost and risk without much benefit.

Inside the working loop.

Exact implementations differ, but many agentic systems repeatedly gather information, act and check the result.

01

Receive the goal

Example: "Check these website pages for broken links and layout problems."

02

Inspect what exists

Open relevant files, visit pages or read reports through permitted tools.

03

Take a small action

Run a check, prepare a change or create a draft. Higher-risk actions should require approval.

04

Check and continue

Compare the result with the goal. Fix errors where authorised, or report blockers and stop.

NOT MAGIC AUTONOMY

A model cannot perform an action just by describing it. The tool, permission and runtime must genuinely exist.

04 / TOOL ACCESS

Tools turn words into work.

Capabilities depend on the product and setup. Access to one tool does not imply access to everything else.

READ

Search & retrieve

Find files, read documentation or gather public information. Access still needs to respect privacy and source quality.

WRITE

Edit & prepare

Create a file, update code or fill a draft. Use narrow permissions, backups and review for consequential changes.

EXECUTE

Run & test

Execute a permitted command or test suite, then inspect the output. A successful command is not the same as a successful project.

CONNECT

Apps & APIs

Interact with authorised services, such as a task board. Actions such as sending, paying or publishing need care.

PERMISSIONS FIRST

Start with read-only access where possible. Require confirmation for deletion, publication, spending or irreversible actions.

05 / USE CASES

Three useful agent jobs.

Good agent tasks have clear goals and checks, but also contain steps where the next move depends on what is found.

These are realistic examples, not a claim that every AI product supports them out of the box.

WEB DEVELOPMENT**Review a website build**

Inspect project files, run tests, check navigation and generate a report of issues. Apply fixes only within an agreed scope.

ADMINISTRATIVE WORK**Prepare, do not auto-send**

Read authorised notes, organise follow-ups and draft messages. A person checks recipients and content before anything is sent.

RESEARCH**Build a cited briefing**

Collect sources on a question, compare conflicting claims and produce a summary that distinguishes facts from uncertainty.

The best use cases save repeated effort while leaving risky judgement and irreversible decisions with the user.

06 / HANDOFF PROMPT

Give the agent a real brief.

Planning through discussion still matters. Turn agreed decisions into a self-contained task for a fresh agent session.

A vague instruction like “fix my whole website” is difficult to verify. A bounded brief makes results and limits clearer.

GOAL**Say what success looks like**

“Review the public site for broken navigation, missing assets and major mobile layout issues. Produce a prioritised issue report.”

SCOPE**Define tools and boundaries**

“Use only the local copy and staging site. Do not publish, delete content or change production credentials. Ask before external actions.”

CHECKS**Specify what to return**

“Run available checks, verify changes, list anything untested, and give file paths, test outcomes and next steps.”

Before starting, review the brief and give the agent the actual files and tool permissions it needs.

Keep the human in charge.

An agent may act quickly and repeatedly. Give it enough access to do the job, but no more.

PERMISSIONS

Least privilege

Use the minimum files, accounts and tools needed. Read-only first; make write access explicit and reversible.

APPROVALS

Pause for high-stakes actions

Require a human decision before publishing, sending important messages, deleting data, moving money or changing production systems.

UNTRUSTED INPUT

Beware of instruction injection

A web page or document might contain hostile directions. Treat retrieved content as data, not as orders from the user.

RECORDS

Logs and checkpoints

Track changes, tests, decisions and errors. Good audit trails make mistakes easier to detect and recover from.

IMPORTANT

A prompt saying “be careful” is not a substitute for tool permissions, isolation and real technical controls.

08 / FAILURE MODES

Why agents get stuck.

Long tasks do not always finish cleanly. Recognise the patterns rather than endlessly repeating the same prompt.

LOOPS**Repeating a failed step**

The agent keeps retrying a broken command. Set a retry limit and ask it to report the blocker.

CONTEXT**Losing earlier decisions**

Long sessions can lose track of constraints. Keep a written brief, checkpoints and a clear list of agreed requirements.

FALSE COMPLETION**Claiming success too soon**

A report may say “fixed” when no test passed. Require actual command output, file changes and manual spot-checks.

COST**Unbounded work**

Tools, long conversations and repeated calls consume time and allowance. Set a budget and stop conditions before starting.

GOOD PRACTICE

Checkpoint after each meaningful phase: changed, tested, outstanding, next action.

09 / TRY IT

Your first supervised agent.

Try a low-risk website quality review on a local project. Keep the agent read-only until you trust the workflow.

01 Pick one bounded goal

For example, identify broken internal links on a staging site.

02 Provide the real context

Supply the local folder, relevant URLs and known constraints.

03 Start with safe tools

Allow reading and testing; avoid unnecessary account credentials.

04 Set acceptance criteria

Request a list of checked pages, failures and reproduction steps.

05 Define stop conditions

Stop on repeated errors, missing access or spending limits.

06 Review the evidence

Open a sample of reported issues and verify them yourself.

07 Approve changes separately

Only then authorise fixes, tests and a documented checkpoint.

MAKE IT REPEATABLE

Keep the prompt and checklist. A repeatable small workflow is more valuable than an unbounded “do everything” agent.

10 / KEEP LEARNING

Start small. Stay involved.

The goal is not to hand over every decision. It is to let tools handle repetitive steps while you keep control of outcomes.

A useful agent follows a clear brief, works through permitted tools and shows what happened. Begin with narrow tasks, check the results, and only expand autonomy when the workflow has earned your trust.

REMEMBER THESE THREE THINGS

01 Tools are the difference

The system needs actual capabilities, not merely confident language.

02 A good brief matters

Goals, limits and definition of done guide useful work.

03 Oversight is part of the design

Permission boundaries, checkpoints and approval are essential.

FURTHER READING / REFERENCE

OpenAI / Building agents



OpenAI / A practical guide to building agents



Anthropic / Building effective agents



Explore more at nathm.net