

Creating a skill step by step

From the task you repeat to a ready-to-use SKILL.md file

Gianluca Bonomo · www.gianlu.it · info@gianlu.it



Contents

How to use this handout	3
1. Before writing: choosing the task	3
2. Anatomy of a skill	4
2.1 The header (frontmatter)	4
2.2 The body: the instructions	5
2.3 Supporting files.....	5
3. Three ways to create it in claude.ai.....	5
3.1 The simplest way: Create with Claude	6
3.2 Uploading a ready folder	6
4. Creating a skill in Claude Code.....	7
4.1 Useful header options	7
5. Testing and improving.....	8
5.1 Six rules for a good skill	9
Hands-on exercise	10
Questions that come up in class.....	11
Glossary.....	11
Further reading	11

How to use this handout

This handout gathers, in written and ordered form, everything covered in the lesson. You do not need to memorise it: it is there so you can find things again calmly, later, when the lesson is over and a question is left hanging.

GOAL

Write, install and test your own skill, choosing between guided creation with Claude, the claude.ai editor and Claude Code folders.

What you will be able to do

- Recognize a task worth turning into a skill.
- Write name, description and instructions of a SKILL.md file.
- Create the skill with Claude, with the editor or in a Claude Code folder.
- Test it with requests that should and should not trigger it.
- Improve it over time with small corrections.

1. Before writing: choosing the task

The best skill comes from a job you have already done well at least three times.

Do not start from zero: open an old chat in which Claude produced exactly what you wanted and ask yourself **what did I have to explain to it?** Those explanations (tone, structure, checks) are the content of the skill.

Is it worth it?

Repeated

You do it often

Regular

Same rules

Model

Typical result

Shared

Others need it

Questions to decide whether a task deserves a skill.

Question	If the answer is yes
Do I repeat it often?	Ideal candidate
Do I always have the same rules?	Put them in the skill
Is there a model of the result?	Attach it as a supporting file
Does the same procedure serve several people?	Share it
Is it an action with real effects (send, pay)?	Always require confirmation

How to choose.

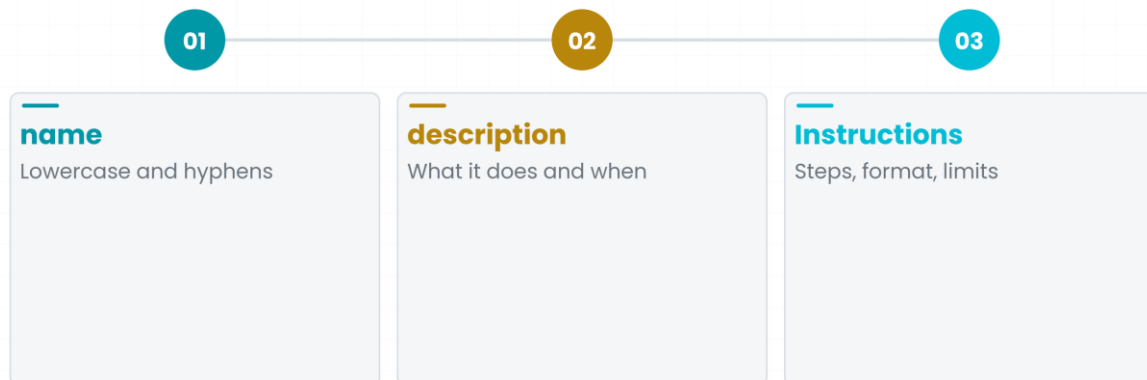
IN PLAIN WORDS

A skill = what you would explain to a new colleague on day one, written once and for all.

2. Anatomy of a skill

A skill is a folder with one required file, SKILL.md, and everything else optional.

Anatomy of SKILL.md



The three parts of a SKILL.md file.

2.1 The header (frontmatter)

At the top of the file, between two lines of dashes, go two required fields:

Field	Rules (official documentation)
name	Maximum 64 characters; only lowercase letters, numbers and hyphens; no XML tags; cannot contain the reserved words «anthropic» and «claude»
description	Required, maximum 1,024 characters; must say what the skill does and when to use it; no XML tags

The two required fields.

2.2 The body: the instructions

After the header write in Markdown, step by step: what to do, in what order, with what format, what to avoid. Use direct verbs («Read», «List», «Ask for confirmation before sending») and a complete example of the expected result.

2.3 Supporting files

Templates, price lists, examples and scripts live in separate files (for example `template.docx`, `examples.md`, `scripts/check.py`) called from the body. Claude opens them only if needed.

EXAMPLE: MEETING-MINUTES

name: meeting-minutes

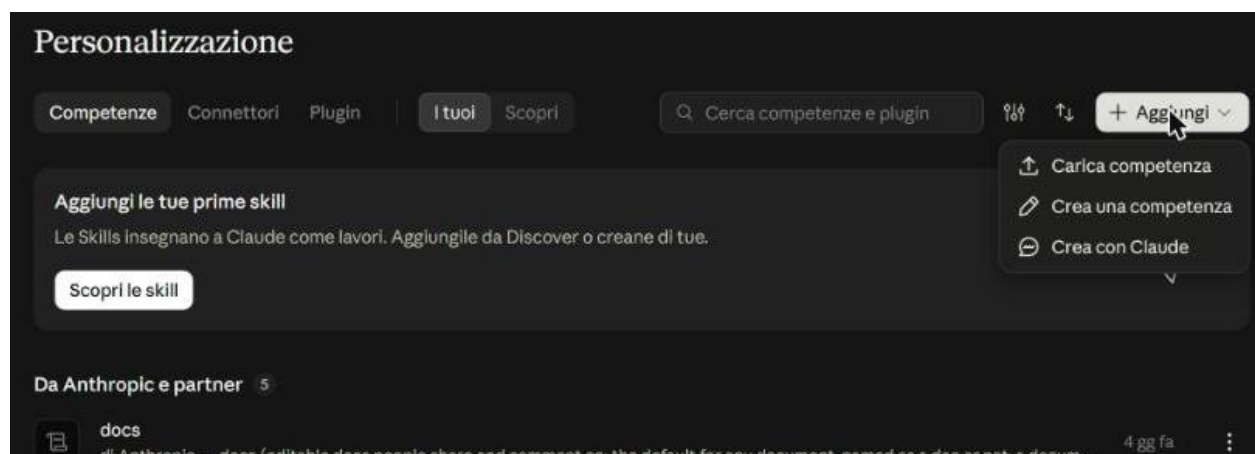
description: Turns notes or the transcript of a meeting into minutes with decisions, actions and deadlines. Use it when the user asks for minutes, a report or the key points of a meeting.

Meeting minutes

1. Read the notes and identify participants, date, topic.
2. Write at most 5 decisions, one line each.
3. List the actions in a table: who, what, by when.
4. Flag ambiguous points instead of inventing them.
5. Finish with a one-sentence summary.

3. Three ways to create it in claude.ai

In Customize, the Add button offers three paths.



Skills, Add (Italian interface): Upload skill, Create a skill, Create with Claude.

Entry	When it suits
Create with Claude	The simplest: Claude interviews you and writes the skill (it uses Anthropic's «skill-creator» skill)
Create a skill	You write name, description and instructions yourself in an editor
Upload skill	You already have a folder ready: you upload it as a compressed file (zip)

The three paths of the Add button.

3.1 The simplest way: Create with Claude

1. Open **Customize, Skills**, press **Add** and choose **Create with Claude**.
2. Tell it about the job: what you want to get, an example of a good result, the rules never to break.
3. Answer Claude's questions.
4. Review the draft of the skill and ask for corrections.
5. Save and turn on the skill; then try it in a new chat.

3.2 Uploading a ready folder

Compress the skill's folder (with the SKILL.md file inside) into a zip archive and choose **Upload skill**. Custom skills require a paid plan with code execution on.

CAREFUL

Do not write passwords, keys or customer data in the skill: the text can be read by anyone with access to the skill and ends up in the conversation context.

4. Creating a skill in Claude Code

In Claude Code a skill is simply a folder on your computer.

Type	Path	Available
Personal	~/.claude/skills/name/SKILL.md	In all your projects
Project	.claude/skills/name/SKILL.md	Only in that repository
Plugin	the plugin's skills folder	Where the plugin is enabled

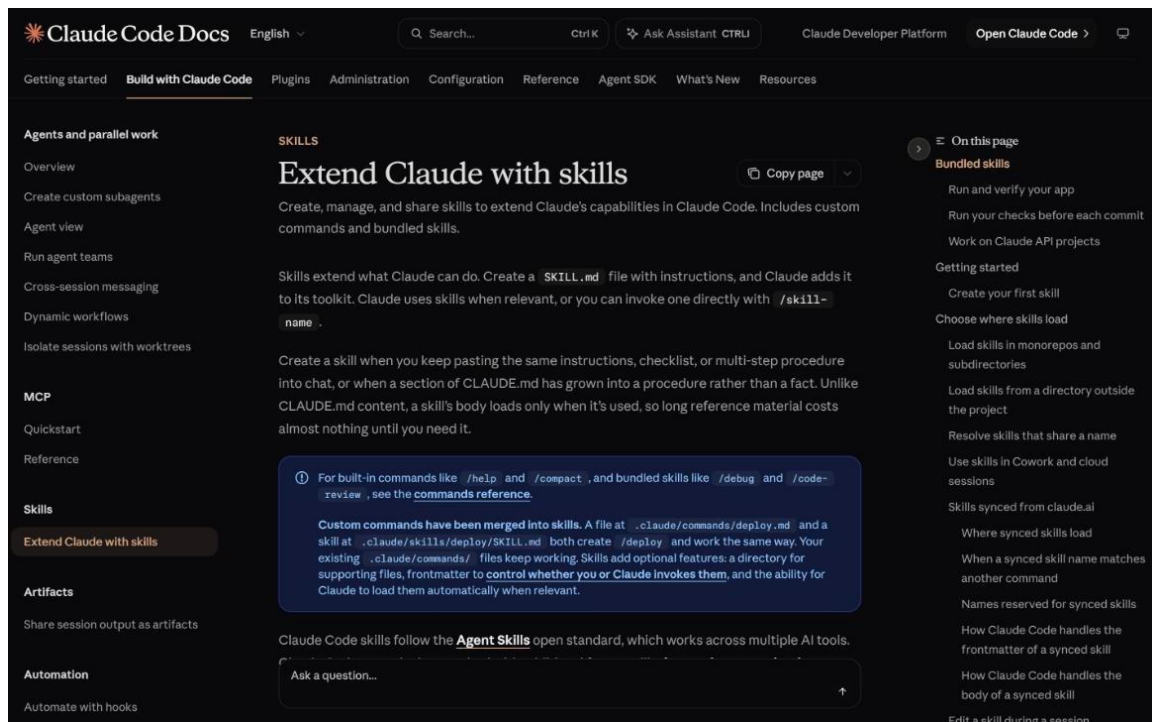
Where skills live in Claude Code.

To create one, a folder with a SKILL.md file is enough. Then you call it with /name or Claude uses it by itself when the request matches. On Windows ~ corresponds to your user folder (C:\Users\yourname).

4.1 Useful header options

Field	What it is for
disable-model-invocation: true	Only you can start it: for actions with effects (publish, send, commit)
allowed-tools	Pre-approves some tools to avoid constant confirmation requests
arguments and \$ARGUMENTS	Passes parameters: /fix-ticket 123
context: fork	Runs the skill in a separate context (subagent)

Some fields documented for Claude Code (October 2026).



The official guide to Claude Code skills.

SKILL WITH AN ARGUMENT

name: fix-ticket

description: Fix a GitHub ticket. Use it when the user gives a ticket number.

disable-model-invocation: true

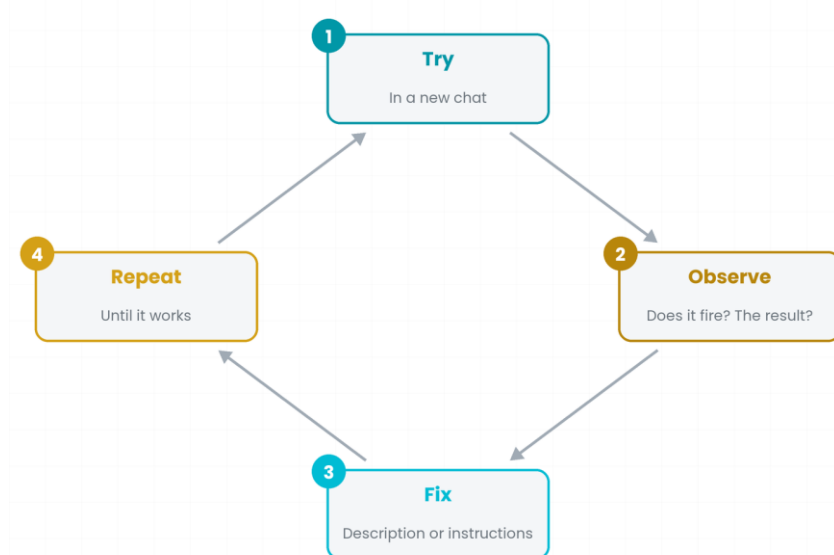
Fix GitHub ticket \$ARGUMENTS:

1. Read the ticket description.
2. Identify the files involved.
3. Implement the fix and write a test.
4. Run the tests and summarize the changes.

5. Testing and improving

A skill is not finished when you save it: it is finished when it works in a clean chat.

Testing



The skill testing cycle.

1. **New chat:** try in a clean conversation, without the context in which you wrote it.
2. **Requests that should trigger it:** phrased in different ways (even in your everyday words).
3. **Requests that should not trigger it:** check that it stays off.
4. **Read the output:** is a step missing? Is the tone right? Does the format match?
5. **Fix:** in the description if it does not fire, in the instructions if the result is wrong.

5.1 Six rules for a good skill

- **One job** per skill: better three small skills than one huge one.
- **Concrete description**, with the words the user would use.
- **Short instructions** in the body; details in supporting files.
- **An example** of the expected result.
- **State the limits:** what it must not do and when to ask for confirmation.
- **Test** in a new session and review now and then.

CAREFUL

Skills do not sync between claude.ai, Claude Code and the API: if you use them in several places, upload them to each and keep a master copy of the folder.

Hands-on exercise

Thirty minutes to create and test your first skill, «meeting-minutes».

1. Open Customize, Skills, Add, Create with Claude.
2. Describe the task: «Turn meeting notes into minutes with decisions, actions (who, what, by when) and points to clarify».
3. Paste an example of made-up notes and an example of a well-made set of minutes.
4. Review the proposed skill: does the description say what it does and when to use it?
5. Save and turn it on; open a new chat and paste other test notes.
6. Try a request that does not fit (for example a recipe) and check it does not fire.
7. Fix one thing you do not like and repeat.

WATCH OUT

Is the name lowercase with hyphens?

Does the description contain both what it does and when to use it?

Did you try it in a new chat?

Did you avoid real data in the test texts?

Questions that come up in class

Do I need to know how to program to create a skill?

No: a skill is a text in Markdown. Code (scripts) is optional.

How long should it be?

As long as needed: the core instructions fit nicely in a page or two, with details in separate files.

Why does my skill not fire?

Almost always because of the description: rewrite it with the words you really use and say when to use it. As a test, name the skill in the request.

Can I share it with colleagues?

In claude.ai everyone uploads their own copy; in Claude Code you can put it in the project's repository or in a plugin.

Can I have Claude update it?

Yes: ask it to improve it by reading the examples that did not work, then review the changes yourself.

Glossary

Term	What it means
Frontmatter	The header at the top of the file, between two dash lines, with name and summary text.
Markdown	A simple text format with symbols for headings and lists.
Supporting file	Templates, examples or programs called from the main instructions.
Trigger	The request that makes the capability start, decided by its summary text.
Zip	A compressed archive used to upload a capability to claude.ai.

Further reading

Anthropic, Agent Skills overview: structure and limits of name and description - platform.claude.com/docs, October 2026

Anthropic, Extend Claude with skills (code.claude.com/docs/en/skills) - paths and header fields

claude.ai interface, Customize, Add - screenshot of 3 October 2026

WATCH OUT

Limits and fields verified on 3 October 2026 against the official documentation; check your plan's requirements.