

Claude Code: the assistant that programs

Installation, first assignment, permissions, CLAUDE.md and good habits

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



Contents

How to use this handout	3
1. What Claude Code is	3
1.1 Who can use it.....	3
1.2 Where you use it	4
2. Install and get started	4
2.1 On Windows.....	4
2.2 On macOS and Linux.....	5
2.3 The first launch	5
2.4 Without a terminal: web and app	5
3. The method: explore, plan, code, verify	6
3.1 Plan mode.....	6
3.2 Give it a way to verify.....	6
4. Permissions, memory and essential commands	7
4.1 Permission modes.....	7
4.2 CLAUDE.md: the standing instructions.....	7
4.3 Commands to know	8
5. Extending Claude Code	8
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

Understand what Claude Code is, install it, give it a first job on a project and manage it with permissions, standing instructions and a good method.

What you will be able to do

- Choose the right surface: terminal, desktop app, web or editor.
- Install Claude Code and start a session in a project folder.
- Use plan mode, permissions and the essential commands.
- Write a short, useful CLAUDE.md file.
- Apply the method explore, plan, code, verify.

1. What Claude Code is

Claude Code is an assistant that reads your project, edits the files and runs commands, while you watch, correct and decide.

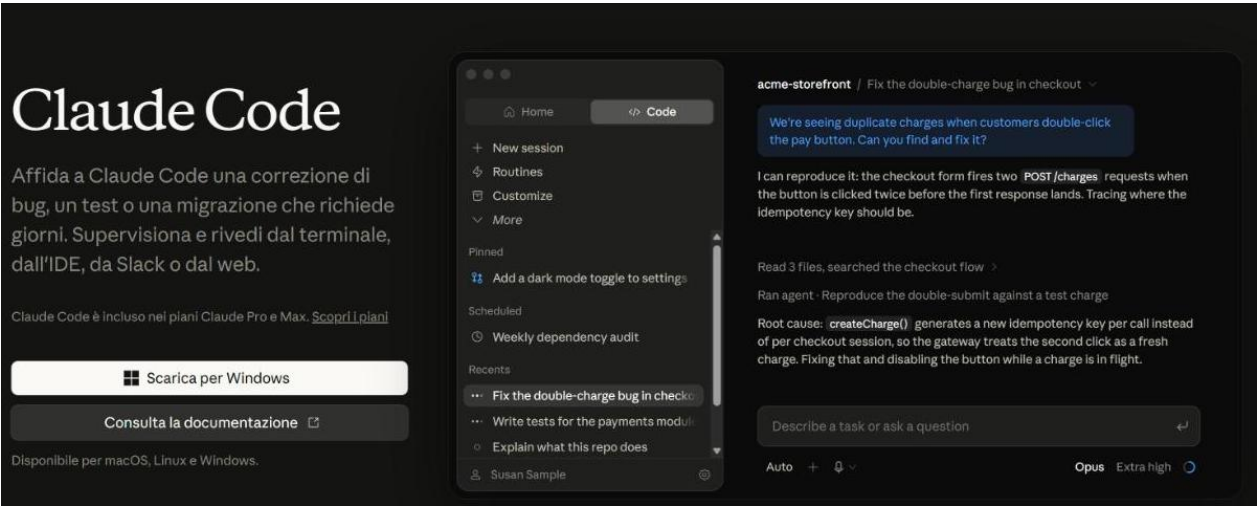
According to the official documentation, Claude Code is an «agentic» coding tool: it reads code, edits files, runs commands and connects to development tools. You describe in plain language what you want (a new feature, a bug fix) and it plans, writes the code across several files and checks that it works.

IN PLAIN WORDS

A chat tells you how to do it. Claude Code does it, directly in your project's files, and shows you every change.

1.1 Who can use it

You need a paid plan: the Italian page of October 2026 says Claude Code is included in the Claude Pro and Max plans, available for macOS, Linux and Windows.



The official Claude Code page (claude.com/product/claude-code, Italian version).

1.2 Where you use it

Surface	How you use it
Terminal	The claude command, in the project folder
Desktop app	The Code tab of the Claude app, with graphical review of changes
Web	claude.ai/code: cloud sessions on GitHub repositories
Editor	Extensions for VS Code and JetBrains
Mobile and Slack	Start and follow tasks from your phone or from Slack too

The Claude Code surfaces. All use the same engine.

2. Install and get started

Installing takes one command; the rest is talking.

2.1 On Windows

Open PowerShell and paste the command shown in the official documentation.
Alternatively use the WinGet package manager.

COMMANDS (WINDOWS)

PowerShell: irm https://claude.ai/install.ps1 | iex

Or WinGet: winget install Anthropic.ClaudeCode

2.2 On macOS and Linux

COMMANDS (MACOS, LINUX)

Terminal: `curl -fsSL https://claude.ai/install.sh | bash`

Or Homebrew: `brew install --cask claude-code`

When the installation finishes, open a **new** terminal and type `claude --version`: if a number appears, it worked. On Windows Git for Windows is also recommended.

2.3 The first launch

1. Enter the project folder (`cd my-project`).
2. Type `claude` and press Enter.
3. At first launch sign in with your Claude account.
4. Describe what you want, for example: «explain how this project is organized».

2.4 Without a terminal: web and app

If the command line is not for you, open the **Code** tab in the desktop app or go to `claude.ai/code`. In the web version Claude works on a cloud copy of your GitHub repositories and returns a change proposal for you to review.



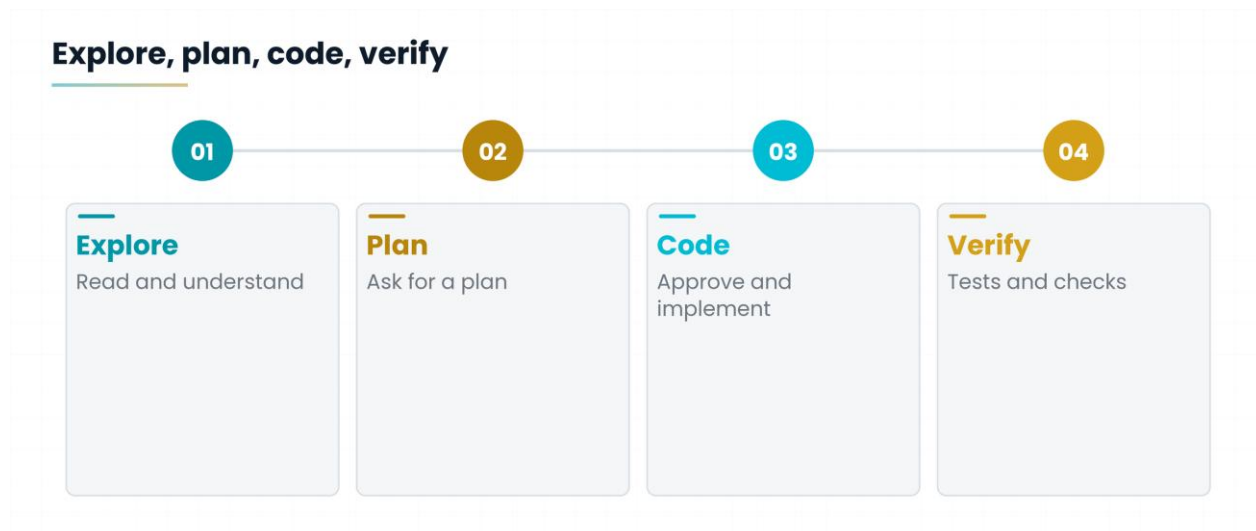
Claude Code on the web (Italian interface): Claude clones the repository into a secure cloud environment.

CAREFUL

Work on a copy or on a test branch. For a first experiment use a folder without important data.

3. The method: explore, plan, code, verify

The best results come when you separate understanding from doing.



The method recommended by the official documentation.

1. **Explore:** ask Claude to read the files and explain how things work.
2. **Plan:** ask for a detailed plan before changing anything.
3. **Code:** approve the plan and have it implemented.
4. **Verify:** have it run tests or checks and look at the result.

3.1 Plan mode

Pressing **Shift + Tab** in the terminal switches to plan mode: Claude reads and reasons but changes nothing. When the plan convinces you, you approve it and Claude carries it out. For trivial changes (a typo, a variable to rename) you can skip the plan.

3.2 Give it a way to verify

The most important advice in the documentation: give Claude something it can check by itself, like a set of tests, building the project or a screenshot of how it should look. Without a check, the only one verifying is you.

BEFORE AND AFTER

Before: «Make a function that validates emails.»

After: «Write the validateEmail function. Examples: user@example.com is valid, user@.com is not. After writing it run the tests and fix until they pass.»

4. Permissions, memory and essential commands

Claude Code can edit files and run commands: you decide how much to trust it.

4.1 Permission modes

Mode	What happens
Manual (default)	It can only read: it asks for confirmation for changes and commands
Accept edits	Applies file changes by itself; asks for the rest
Plan	Reads and proposes, without changing anything
Auto	A second model reviews the actions and blocks risky ones

The main modes (official documentation, October 2026): switch with Shift + Tab.

4.2 CLAUDE.md: the standing instructions

CLAUDE.md is a text file in the project folder that Claude reads at the start of every session. It holds the commands to run tests, style rules and the project's quirks. You generate it with the `/init` command and keep it **short**: a file that is too long is partly ignored.

EXAMPLE CLAUDE.MD

```
# Style
- Use ES modules (import/export), not require

# Work
- After changes run the type check
- Prefer running a single test, not the whole suite
```

4.3 Commands to know

Command	What it is for
/init	Creates a starter CLAUDE.md
/clear	Resets the context between different tasks
/compact	Summarizes the conversation to free up space
/rewind or Esc Esc	Goes back to an earlier point (conversation and code)
/permissions	Manages pre-approved permissions
/hooks	Shows the configured automations

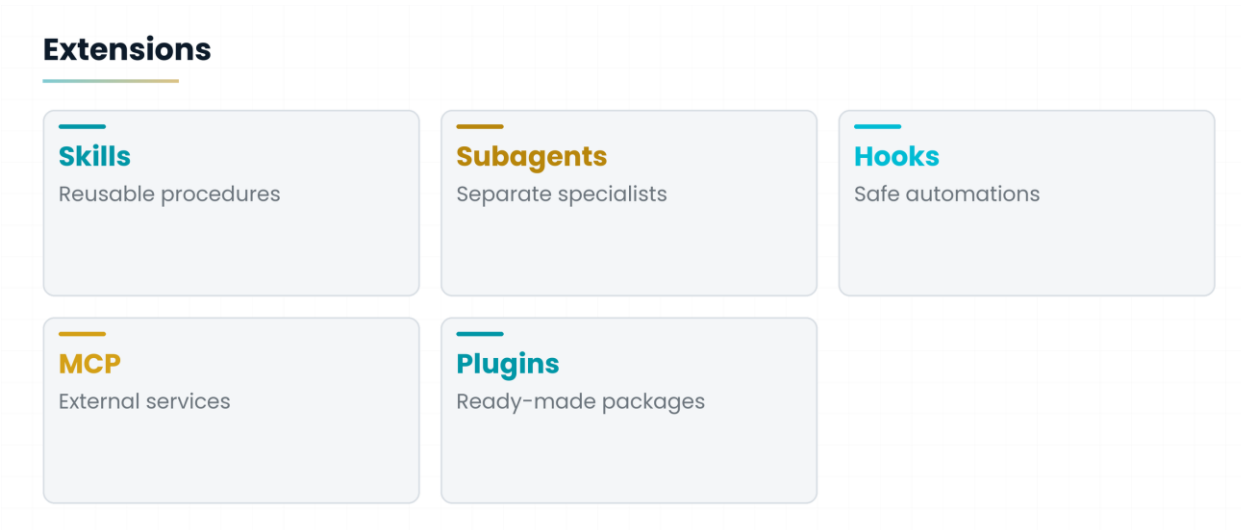
Essential commands.

CONTEXT IS THE KEY RESOURCE

The fuller the session, the more Claude forgets and errs. Use /clear between different tasks and, after two failed corrections on the same problem, start over with a more precise request.

5. Extending Claude Code

When the basics work, you can add tools: each has a precise purpose.

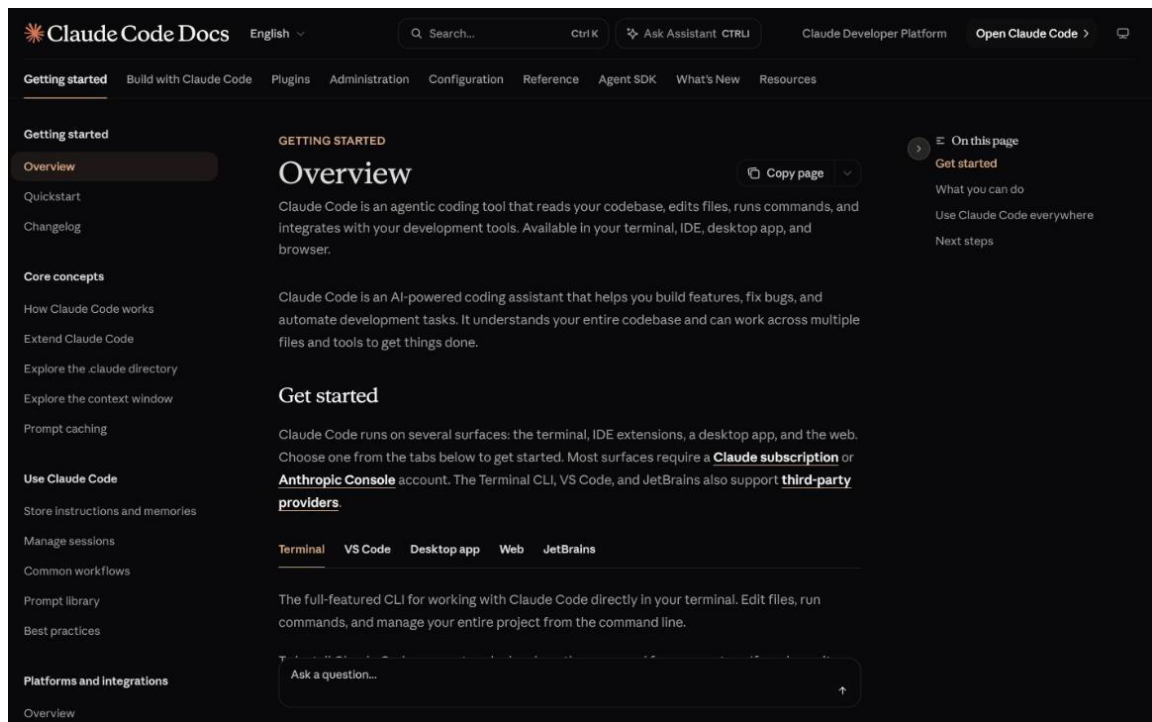


Five ways to extend Claude Code.

- **Skills:** reusable procedures, called with /name (chapters 5 and 6).
- **Subagents:** specialized assistants with their own context, for research or reviews.

- **Hooks:** commands run automatically before or after an action (for example formatting the code on every edit).
- **MCP:** links to external services, added with `claude mcp add` (chapter 7).
- **Plugins:** packages that gather skills, hooks, subagents and MCP.

The official documentation is very rich: it has a guide site (code.claude.com/docs) and a free course, Claude Code 101, at academy.claude.com.



The Claude Code documentation (code.claude.com/docs).

Hands-on exercise

Thirty minutes to install Claude Code and have a small project explained to you.

1. Create a folder «code-test» with two or three arbitrary text or code files.
2. Install Claude Code with the command for your system and check with `claude --version`.
3. Enter the folder, start claude and sign in.
4. Ask: «Explain what these files contain and propose a plan to improve them. Do not change anything.»
5. Run `/init`, read the CLAUDE.md created and shorten it to 5-8 lines.
6. Try `/clear` and repeat a question.

WATCH OUT

Does `claude --version` show a number?

Did you ask for a plan before changing anything?

Is the CLAUDE.md short and specific?

Questions that come up in class

Do I need to know how to program?

Not to start exploring a project, but to review changes it helps to know the basics. Claude Code remains a tool for people who develop.

What is the difference from the chat?

The chat answers with text. Claude Code acts on your files and commands, with your permission.

Can it break my project?

It can edit files, which is why you work on branches or copies and check permissions. Checkpoints (/rewind) exist but do not replace git.

How much does it cost?

It is included in the Pro and Max plans; heavy use can exhaust the limit. See chapter 8.

Does it work on Windows?

Yes: install with PowerShell or WinGet; Git for Windows is recommended.

Glossary

Term	What it means
CLAUDE.md	A standing instructions file for Claude Code.
Hook	A command run automatically before or after a Claude action.
Plan mode	A mode in which Claude reads and proposes without changing anything.
Repository	The folder of a software job, with its change history (git).
Subagent	A specialized assistant with a separate context.
Terminal	A window where you type text commands.

Further reading

Anthropic, Claude Code Docs: Overview and Best practices (code.claude.com/docs) - accessed 3 October 2026

Anthropic, Claude Code product page (claude.com/product/claude-code) - accessed 3 October 2026

Anthropic, Claude Code Docs: Permission modes - October 2026

WATCH OUT

Commands and modes verified on 3 October 2026 against the official documentation; Claude Code is updated often.