
BYKOV-BRETT ENTERPRISES · WORKSHOP

Getting Claude Set Up in VS Code

A hands-on, follow-along workshop. Install the assistant, sign in once, and learn to use it safely.

90 minutes

Beginner-friendly

Windows · macOS · Linux



Turn Intent Into Impact.



From “I’ve heard of it” to “it’s running on my laptop”

Claude is an AI assistant that can live right inside VS Code, the free code editor used by millions of professionals. This session takes a complete beginner from nothing installed to a working, signed-in assistant they can actually use.

It is built for non-technical people. No coding experience is assumed. Everyone follows along on their own laptop, one step at a time, and leaves with the tool set up and a clear sense of how to use it responsibly.

VS Code is the same application on Windows, macOS and Linux, so one set of instructions works for the whole room. The few differences that exist are small, and they are called out as we go.

What every attendee leaves with

- ✓ The genuine Claude extension installed
- ✓ Signed in and confirmed working
- ✓ A first prompt and an accepted change
- ✓ The safety rules to use it at work

This tool acts on your real computer

Claude doesn't just chat. In the folder you open it can create, change and delete files, and it can run commands in your terminal. Cover this before anyone installs or runs anything.



It acts on your real machine

In the folder you open, Claude can create, edit and delete files and run terminal commands. These are real changes to your computer, not a safe sandbox.



You stay in control — through the permission mode

In the default mode Claude asks before each change. That safety depends on which permission mode you're in (see the Permission settings page). Don't loosen it on unfamiliar work.



Open only what you can afford to change

Work in a folder that's backed up or under version control. Don't point Claude at critical files, system folders, or your whole drive.



Install only what you're allowed to

If your device is managed by an employer, follow IT policy and get approval before installing. You install and run at your own risk.



Your data is your responsibility

Don't enter confidential, personal or client information — your inputs are sent to a third party. See the safety rules later in this pack.



AI can be wrong · costs are yours · you accept the risk

Check anything important before relying on it. Claude Pro and Max are paid — charges sit on your account. By taking part you accept responsibility for your own device, accounts and data.



The permission mode is your main safety control — it decides what Claude can do to your files and terminal without asking. The next page explains each mode.

What we'll achieve, and what you need first

By the end, you can...

- 1 Install the genuine Claude extension by Anthropic, and spot the imitations.
- 2 Sign in once through the browser and confirm Claude is connected.
- 3 Ask Claude about your work, and safely accept or reject a suggested change.
- 4 Open the built-in terminal and run Claude, seeing it is the same tool.
- 5 Use Claude safely at work, and know exactly where to get help.

BEFORE YOU START — sent the day before

- ✓ A laptop you can install software on (admin rights).
- ✓ VS Code installed, version 1.98 or newer.
- ✓ A Claude Pro or Max account, logged in to your browser. The free plan has no Claude Code.
- ✓ A reliable internet connection.

What to bring, and what each must contain

Slide deck

A title slide, the running-order timings, and the five safety rules. Plus screenshots of: the Extensions view, the Spark icon, the browser sign-in, and an accepted diff. Few words, big visuals.

Pre-flight checklist

Sent the day before as tick items: VS Code 1.98+ installed; signed in to a Claude Pro or Max account in the browser; admin rights to install; a working internet connection.

Sample project folder

The README.md, menu.txt and optional hello.py from the next page. Zipped to share, or built live in two minutes. Fictional content only.

Links handout

A one-pager with the official Claude documentation URLs and the Bykov-Brett resource links — the same ones on the Resources page at the back.

Troubleshooting sheet

A printout of the five common snags and their fixes, ready to hand out during the troubleshooting clinic (Activity 9).

Projector or shared screen

To demo each step live — install, sign-in, first prompt — at a size the whole room can follow. Test screen-share before you start.

Prepare your sample project

Activities 5, 6 and 8 need a tiny, throwaway project everyone can open. Create a folder (e.g. “claude-workshop-demo”) with these 2–3 small, fictional files. Zip it to share, or have the room build it live in two minutes.

README.md

REQUIRED

A few lines describing a make-believe project. For example: “Sunrise Café — a simple menu site. A demo project for the Claude workshop.” Gives Claude something to summarise.

menu.txt

REQUIRED

A short, fictional list with a little structure — say five drinks and their prices. Makes “explain what this file does” produce a satisfying answer.

hello.py

OPTIONAL

Five lines with one simple function, e.g. a greet(name) that prints a hello. Lets the “add a comment” diff demo in Activity 6 work nicely.

KEEP IT SAFE

Everything fictional and generic — no real names, clients, credentials or company data. Building it from scratch is itself a live demonstration of “open only what you’re comfortable sharing.”

The 90-minute flow, ten activities

Total: 90 min

#	ACTIVITY	FORMAT	TIME	MATERIALS	FACILITATOR FOCUS
1	Welcome & framing	Plenary	9 min	Slides	Set the tone — “an assistant inside your editor”. Flag that safety comes later.
2	Pre-flight check	Guided	7 min	Checklist	Catch blockers early: free plan, old version, no admin rights.
3	Install the extension	Hands-on	11 min	Projector	Find the real Anthropic one; warn off the look-alikes.
4	Sign in & confirm	Hands-on	9 min	—	Walk the one-time browser sign-in; reassure when it pops open.
5	First prompts	Hands-on	9 min	Sample project	Make sure everyone gets one successful reply.
6	Accept or reject a change	Hands-on	10 min	Sample project	Teach “nothing changes until you accept”. Save the file after accepting.
7	Using Claude safely	Plenary + discussion	12 min	Slides	The five safety rules: what it sees, the permission modes, what not to paste.
8	A taste of the terminal	Demo + hands-on	7 min	—	Same tool, second face. Per-OS callout. Keep it light.
9	Troubleshooting clinic	Plenary + circulate	9 min	Troubleshoot sheet	Work the five common snags; rescue anyone still stuck.
10	Wrap & next steps	Plenary	7 min	Links handout	Point to official docs + BBE resources; give one clear next action.

Activities 1–3 · Welcome, pre-flight, install

1 Welcome & framing 9 min

SETUP

Slides up. Quick room intro.

RUN

- Ask: “who has used an AI chatbot before?” — set a friendly tone.
- Explain: Claude can live inside VS Code; it chats, and it can read or change files you allow.
- Tell them it’s hands-on, and that a proper safety section is coming.

WHAT GOOD LOOKS LIKE

Everyone knows what we’re building, and that nothing risky happens yet.

2 Pre-flight check 7 min

SETUP

Put the checklist on screen.

RUN

- Each person confirms VS Code opens and Help → About shows 1.98 or newer.
- Confirm they can sign in to claude.com with a Pro or Max account.
- Confirm admin rights to install. Pair anyone blocked with a neighbour.

WHAT GOOD LOOKS LIKE

Blockers surfaced now, not mid-install. Nobody stuck in silence.

3 Install the extension 11 min

SETUP

Project the Extensions view.

RUN

- Open Extensions: Ctrl/Cmd + Shift + X.
- Search “Claude Code”; choose the one published by Anthropic (verified, millions of installs).
- Click Install; reload the window if prompted.
- Warn: ignore look-alikes that borrow the Claude name.

WHAT GOOD LOOKS LIKE

Extension shows Installed; the Spark icon appears once a file is open.

Activities 4–6 · Sign in, first prompts, the diff

4 Sign in & confirm 9 min

SETUP

Have a file open so the Spark icon shows.

RUN

- Click the Spark icon (top-right), or the Claude Code item in the status bar.
- Click Sign in; a browser tab opens for a one-time sign-in.
- Choose the Pro or Max account; return to VS Code.
- If the browser doesn't open, use the copy-link option and paste it.

WHAT GOOD LOOKS LIKE

The panel shows a signed-in state. No need to log in again.

5 First prompts 9 min

SETUP

Open the sample project; everyone opens a file.

RUN

- In the panel, type: “explain what this file does”.
- Read the reply together; point out it's about their file.
- Try one more: “summarise what's in this folder”.
- Celebrate the first successful reply — this is the win.

WHAT GOOD LOOKS LIKE

Everyone has seen Claude respond about their own files.

6 Accept or reject a change 10 min

SETUP

Open a simple file in the sample project.

RUN

- Ask: “add a friendly comment to the top of this file”.
- Claude shows a diff — a side-by-side of the change.
- Review it together, then click Accept (or Reject).
- Note the file is now unsaved; press Ctrl/Cmd + S to save.

WHAT GOOD LOOKS LIKE

Learners control every change: review first, then save.

Five rules for using Claude safely at work

This is the part that decides whether people use Claude confidently or avoid it. Walk it slowly, and invite questions.

1

Know what Claude can see

The folder and files you open, plus what you type, are sent to Anthropic to generate answers. Open only what you're comfortable sharing.

2

It asks before it acts

Claude proposes a change and waits for you. Stay on "ask" mode while learning; never leave auto-accept running on unfamiliar work.

3

Never paste secrets

Passwords, API keys, client data and personal details do not belong in prompts. Treat it like emailing an outside company.

4

Know the privacy setting

On Pro and Max, your chats are not used to train the model unless you turn that on. Check the setting and follow Anthropic's privacy policy.

5

Follow your workplace's AI policy

Use approved tools, keep sensitive data off-device unless allowed, and ask before opening confidential projects.



DISCUSS IN PAIRS

Name one file or piece of information you should never open or paste into Claude at work.

Who's in charge: the permission modes

One setting decides what Claude can do to your files and terminal without asking. Find it at the bottom of the Claude panel. Walk through the four people who will actually meet.

Ask before edits

the default

RECOMMENDED

Reads your files, but asks before every file change or terminal command.

Plan mode

look before you leap

SAFE

Reads and writes a plan only. Changes nothing on your machine until you approve it.

Edit automatically

faster, less review

WITH CARE

Creates and edits files without asking (and basic file moves). Still asks before other terminal commands.

Bypass permissions

no safety net

DANGER

Does everything with no checks. One mistake can delete your project. Isolated VMs or containers only.

HOW TO SWITCH

In VS Code, click the mode shown at the bottom of the Claude panel. In the terminal, press Shift + Tab to cycle.

MADE A MISTAKE?

Press Esc twice to undo the last change, or just ask Claude to undo it. For today, stay on Ask before edits.

Activities 8–10 · Terminal, troubleshooting, wrap

8 A taste of the terminal

7 min

SETUP

Close the panel; show the built-in terminal.

RUN

- Open the terminal: View → Terminal, or Ctrl/Cmd + ` (backtick).
- Type claude and press Enter.
- Ask: “what does this project do?” — same assistant, in text.
- Type exit to leave.

PER-OS NOTE

Per OS: the extension already bundles the tool, so today just run claude. To install it on its own later — macOS/Linux use the curl script, Windows uses the PowerShell script.

WHAT GOOD LOOKS LIKE

Learners see the panel and terminal are two faces of one tool.

9 Troubleshooting clinic

9 min

SETUP

Put the troubleshooting sheet (next page) on screen.

RUN

- Ask who hit which snag; fix them live.
- Work through the five common snags together.
- Pair anyone who’s finished with someone still stuck.

WHAT GOOD LOOKS LIKE

Everyone is installed, signed in, and has run a prompt.

10 Wrap & next steps

7 min

SETUP

Show the links handout.

RUN

- Recap the five outcomes from the start.
- Point to the official docs and the Bykov-Brett resources.
- Set one next action: “tomorrow, ask Claude to explain one file from your real work.”
- Take final questions.

WHAT GOOD LOOKS LIKE

Everyone leaves with a next step and where to get help.

The five snags people actually hit

 The Spark icon is missing	 Open a file first — the icon needs one. Check VS Code is 1.98+. Reload via Command Palette → “Developer: Reload Window”.
 Looks like the wrong extension	 It must be published by Anthropic. Remove look-alikes, and disable other AI extensions like Cline or Continue that can clash.
 Sign-in won’t complete	 Use the copy-link option and paste it in a browser. Check the Pro or Max subscription is active. Retry quickly — the code expires fast.
 Old VS Code, or the free plan	 Update VS Code (Help → Check for Updates). The free Claude plan has no Claude Code — you need Pro or Max.
 Behind a corporate network	 Install or sign-in may be blocked. Ask IT about approved access, or follow Anthropic’s official network setup docs.

Resources to share with the room

Official Claude documentation

From Anthropic — the source of truth.

-  **Quickstart**
code.claude.com/docs/en/quickstart
-  **Use Claude Code in VS Code**
code.claude.com/docs/en/vs-code
-  **Advanced setup, all platforms**
code.claude.com/docs/en/setup
-  **Install troubleshooting**
code.claude.com/docs/en/troubleshoot-install
-  **The extension on the Marketplace**
marketplace.visualstudio.com — “Claude Code” by Anthropic

Keep learning with Bykov-Brett

Free downloadable guides from this workshop’s author.

-  **Start Here: How to Actually Use AI**
bykovbrett.net/download/start-here
-  **Intent-Driven Design Fundamentals**
bykovbrett.net/download/intent-driven-design-fundamentals
-  **Vibe Coding in Production**
bykovbrett.net/download/vibe-coding-in-production
-  **All guides & resources**
bykovbrett.net/resources

You're set up. Now make it a habit.

You've installed Claude, signed in, run your first prompts, and learned to use it safely. The rest is practice.

→ Tomorrow: ask Claude to explain one file from your real work.

