



Day One: get Claude Code running

From nothing to talking to Claude in about ten minutes. No coding.

Get the Pro plan

The free plan does not include Claude Code. Pro is the cheapest plan that does. Sign up at claude.ai.

Open your terminal

Mac: press Command + Space, type Terminal, press Enter. Windows: press the Windows key, type PowerShell, press Enter.

Paste one line (Mac)

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

Paste it, press Enter, then wait. It shows nothing for a bit while it downloads. That is normal.

Paste one line (Windows)

```
> irm https://claude.ai/install.ps1 |  
iex
```

Same idea in PowerShell. Paste, press Enter, wait.

Type one word

```
> claude
```

Close the window, open a new one, then type this. Says command not found? Close every terminal window, open a fresh one, try again.

Say hi

```
> I run a small business. What can you  
help me with first?
```

Plain English, like texting a person. It logs you in the first time.

Read before you say yes

It asks before it changes anything. Read the ask, then answer. You are always the boss.

Stuck on an error?

```
> What does this error mean, in plain  
English? How do I fix it?
```

Copy the error and paste it right back in. It is great at fixing its own setup.



10 sentences to try on day one

Copy any of these. Change the words to fit your business.

Look around

- > What files are on my desktop?
- > Summarize this PDF in five bullets:

For the PDF, drag the file into the window. That hands it to Claude.

Do the busywork

- > Make a to-do list from these notes:
- > Rename the photos in this folder by the date they were taken

Tidy up

- > Turn this messy list into a clean table:
- > Sort my downloads folder into folders by type

Write the first draft

- > Write a friendly reply to this message. Do not send it:
- > Give me three caption ideas for this photo

Ask about your business

- > What can you help me with in my business?
- > Ask me five questions about my business, then tell me where to start

Check its work

- > Show me what you did and what you were unsure about

Always end with this one. It tells you what to double check.



The keys and commands you will actually use

Ten things. That is all you need to drive Claude Code.

claude

Type it in your terminal to start.

claude -c

Pick up your last conversation where you left off.

Esc

Stop it in the middle of a task. Nothing bad happens.

Esc Esc

Press it twice to go back and change something you said.

Up arrow

Bring back your last message so you can resend or fix it.

Shift + Tab

Switch modes. Plan mode makes a plan first and changes nothing.

Drag a file in

Drop any file on the window to hand it to Claude.

/clear

Fresh start. Use it when you switch to a new topic.

/help

Shows every command if you forget one.

/exit

Quit. Pressing Ctrl + C twice works too.



You are always the boss: 6 safety rules

Claude does the work. You stay in charge of anything that matters.

Say this once at the start

> Before you change, send, post or delete anything, ask me first

It will check with you before anything big.

1. Read the ask

It asks permission before running commands or changing files. Read it. If you are not sure, say no and ask what it means.

2. Escape stops everything

Press Esc any time to stop it mid-task. Then tell it what you want instead.

3. Secrets stay out of the chat

Never paste passwords, card numbers or secret keys. Ask it how to connect a tool safely instead.

4. Drafts first, sends second

> Write a draft reply, but do not send it

Anything that goes to a customer starts as a draft you read.

5. Save before big changes

A save point lets you go back if something breaks. Sheet 06 shows how.

6. Ask what could go wrong

> Is this safe? What could go wrong?

Before anything new, make it think out loud.



The clear directions formula

Vague in, vague out. Four parts get you a good result on the first try.

Vague

> Write an email

You get something generic you have to redo.

Clear

> Write a welcome email for new clients. Friendly, under 100 words, and end with how to book.

Who it is for, what you want, and what good looks like.

1. Who, what and good

Say who it is for, what you want, and what good looks like: length, tone, what it must include.

2. Give an example

> Write it like this example:

Paste one you like. It copies the feel, not the words.

3. Big job? Let it ask

> This is a big job. Ask me questions before you start

It will ask what it needs to know before it starts.

4. Fix it exactly

> Keep the first paragraph, but make the ending shorter and warmer

Say what to keep and what to change. Skip "make it better."

Fill in the blanks

> Make a ___ for ___. It should be ___. Good looks like ___. Here is one I like: ___

Copy this, fill it in, paste it.



Save points: never lose your work

Save points are called git. You never have to learn git. Just ask.

Set it up once

> Set up git in this folder, then save a save point

Do this in any folder you care about, like your website.

Save before big changes

> Save a save point called before the new homepage

Give it a name you will recognize later.

Something broke? Go back

> Something broke. Go back to the last save point

Everything returns to how it was at your last save.

See what changed

> What changed since my last save point? Plain English, please

Great for checking its work before you keep a change.

See your save points

> Show me my save points in plain English

A simple list with dates and names.

Keep a spare copy

> Which of my files are most important to back up?

Save points live on your computer. Keep important files somewhere else too.



Which robot should you hire first?

Pick the one job you would pay to never do again. Start there.



You have never used a terminal

Boot

Free



Graphics take forever

Pixel

\$2.99



Your videos never get edited

Spielbot

\$2.99



You are about to connect email or payments

Bouncer

\$2.99



You never know who still owes you

Penny

\$2.99



Your writing sounds robotic

Echo

\$2.99



You wait on someone to change your website

Dot

\$2.99



Your inbox and bookings eat the day

Tempo

\$2.99



You have forty research tabs open

Scout

\$2.99

Hire one robot for \$2.99, or the whole team for \$14.99. Buy once, keep forever. Start free with Boot at keltysage.com/academy



Stuck? Say **this.**

Copy the one that fits. Plain words fix almost everything.

It showed an error

> That did not work. Here is the error: (paste it). What does it mean in plain English?

Errors look scary. They are just a note about what went wrong, and Claude can read them.

It did the wrong thing

> Undo what you just did

Then say what you wanted instead, with one example.

You are not sure what happened

> What did you just change? Show me in plain English

Ask this any time. You should always know what changed.

It is going too fast

> Slow down. One step at a time, and wait for me after each one

Great for anything new, like connecting a tool for the first time.

The answer is too technical

> Explain it like I have never coded

It will drop the jargon and use everyday words.

It feels too complicated

> Is there an easier way to do this?

There often is. It picks the hard way when it does not know your goal.

It went off in a weird direction

> Stop. Let's make a plan first, then do it

Press Esc first, then type this. Read the plan before you say go.

Before you trust it

> Check your own work and tell me what might be wrong

It is honest about its weak spots when you ask.