

TRAINING

Hands-on AI-Assisted Development

Mechanical and Electrical track Day 2

Intro to vibe coding, prompts, data preparation



Do not redistribute these materials

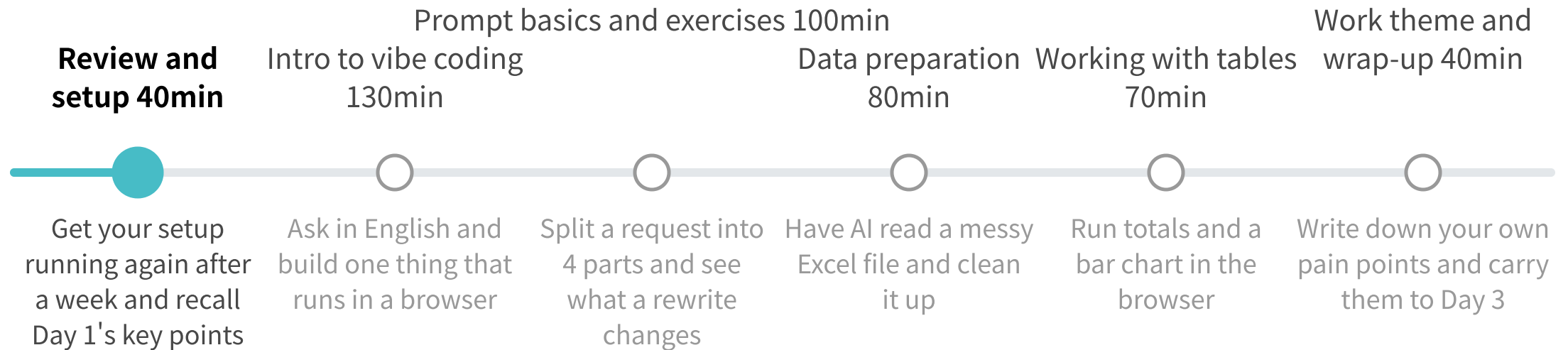
Redistribution of these materials is prohibited.

These materials are copyrighted by Givery, Inc. Use them only for your own learning and work reference. Do not share them outside your company, on social media or with third parties.

During the course, do not take screenshots or recordings that show these materials.

How today runs

Today, **80% of the time is hands-on**



There are two 10-minute breaks: one at the end of the morning and one after the afternoon prompt exercise. The instructor writes the restart time on the board.

01

Day 1 review, setup

Get last week's environment running again and recall Day 1's key points

What we covered on Day 1

Today you work with these 4 points as the basis

How AI works

It is a tool that picks the next words by probability. Check calculations and facts separately, outside the AI.

Strengths and weaknesses

Good at drafts, formatting and rewording. Weak at the latest internal info and numbers that must be exact.

Hiding confidential info

Do not paste client, product, project or partner names. Generalize them to the industry and process level.

Using Copilot

Ask in plain English in the chat on the right of VS Code, then paste the returned code and run it.

We come back to point 3, hiding confidential info, when you describe your work theme this afternoon.

3 tools you use today

Just 3 things: where you write, who you ask, where you run



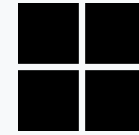
VSCode

Where you create files and write. Today you create index.html here.



GitHub Copilot

Takes your plain-English request on the right of VS Code. It returns code.



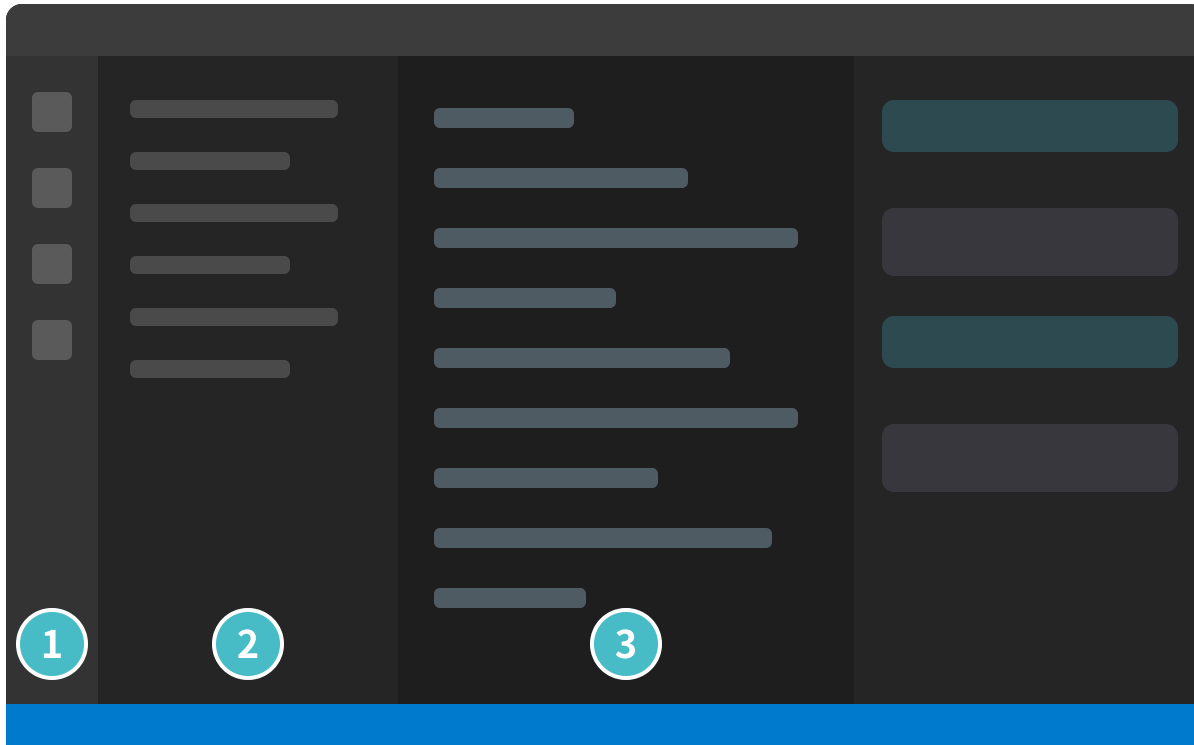
Edge

Open the saved HTML and run it. You check the display here.

You install nothing else. There is no extra setup today.

Today's starting screen

If you can see these 4 areas, you are ready



1 Activity Bar

The bar on the far left. Click the top icon, stacked pages, to show the folder contents.

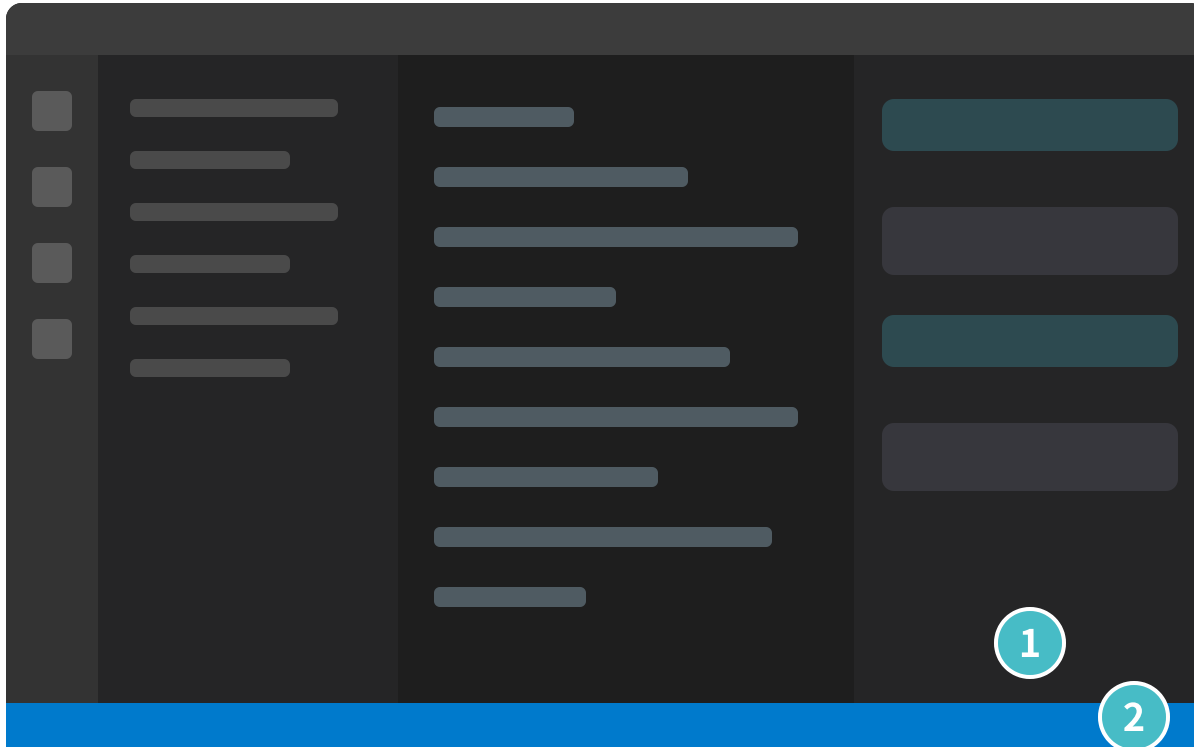
2 Explorer

Contents of the open folder. If it shows tpro-work, today's work folder is open.

3 Editor

Where you write file contents. Paste the code Copilot returns here.

Today's starting screen (continued)



1 Chat view

On the right. Ask here in English. If hidden, click the Copilot icon at top right.

2 Status Bar

The bottom bar. If your GitHub username shows at the bottom left, you are signed in.

The actual starting screen

Check the 4 areas from the previous page **on the real screen**



1 Explorer

Lists the contents of tpro-work. If it does not, open the folder again.

2 Editor

Shows the content of the open file

3 Chat view

If it is not shown, click the chat icon at the top right

4 Input fields

Type all of today's prompts here

Pass these 5 to avoid getting stuck in the next 130 minutes

Scenario

It has been a week since last time. Each time, some people are signed out or cannot open the Chat view. Everyone works on the same screen for the next 130 minutes, so we align first. Fix it now and no exercise time is lost on setup.

Steps

- 1 Start VS Code from the Start menu or the desktop icon.
- 2 Click the Copilot icon at the top right and check that the Chat view opens on the right.
- 3 Look at the account icon at the bottom left. If your GitHub username is not shown, click it and sign in again.

Environment check (continued)

Steps

- 1 In the chat, type "Hello. What can you do today?" and press Enter. Check that a reply comes back in English.
- 2 Create a folder named tpro-work on the desktop, then open it in VS Code with File > Open Folder.

Result

No deliverable. You are done when the tpro-work folder is on your desktop and the VS Code Explorer shows DAY2.

Completion criteria for the environment check

Done when all 4 are in place

Done when

- The Chat view is open on the right
- The account icon at the bottom left shows your GitHub username
- A question typed in English got a reply in English
- The Explorer heading shows DAY2

If you're stuck

- Sign-in does not move on. The browser is open, waiting for approval. Approve in the opened tab. If no approval screen appears, the network is restricted; raise your hand.
- No Copilot icon. Your VS Code may be old. Chat is built in from version 1.116 on.
- You opened only a file, not a folder. Then you lose track of where files are saved. Start again from Step 5.
- No reply. First check the notifications at the bottom right. If it does not recover, you may go on while watching your neighbor's screen.
- Cannot find where you made the tpro-work folder. If you made it somewhere other than the desktop, it is faster to make it again.

Completion criteria for the environment check (continued)

If time allows

- Type "Write HTML code that shows today's date on the screen" in the chat and look over the returned code. It has the same form you use in the morning exercise.
- Compare whether the reply opens in the same tone as when you tried it on Day 1.
- Right-click in the Explorer and look at the menu that appears. In the next section you create a new file from here.

You use one folder

Use tpro-work from the extracted ZIP as your work folder

What is inside

Extract it to the desktop and you get tpro-work. It holds 01_data-loader.html, equipment-data_2026Q1.xlsx, README.txt and a lib folder with xlsx.full.min.js, papaparse.min.js and chart.umd.js.

What you add today

index.html in the morning; prompt.txt, clean.csv, chart.html and theme.txt in the afternoon.

Moving a library

During the afternoon chart session, copy chart.umd.js from the lib folder to the top of tpro-work. This is the only move.

You use this tpro-work folder on Day 3 too. Do not delete it after the course.

Where to check when it fails

Know where to look for each symptom and fix it yourself

Symptom	Where to look	Fix
No Copilot icon	Icons at the top right	Reopen VS Code. If it is older than 1.116, raise your hand.
Asked to sign in	Account at bottom left	Sign in again with GitHub and approve in the browser.
No reply	Bottom-right alerts	May be a network limit. If it persists, tell the instructor.
Last folder missing	File > Open Recent	Create the tpro-work folder again and open it.
Screen in English	Display language	Fine as is. Keep the VS Code display language in English.

From setup check to practice

Choose your own topic. Use your own idea if you have one

What we have done so far

Everyone's VS Code chat now opens and replies in English. tpro-work is on the desktop, apart from the handout folder. You also checked the 4 Day 1 points.



What we do next

In the next 130 minutes, you build one working thing by asking in plain English. You choose the topic; a game is fine. You write no code. Most of the time is hands-on, and the instructor walks around.



Why it matters later

The "say it, run it, point out the gap" loop you learn here is the same approach you use for the afternoon Excel cleanup and the Day 3 mini app. The morning is for getting used to it.



02

Intro to vibe coding

You will be able to ask in plain English and build one thing that runs in the browser

Run it before the design is fixed, then decide after trying it

The usual approach

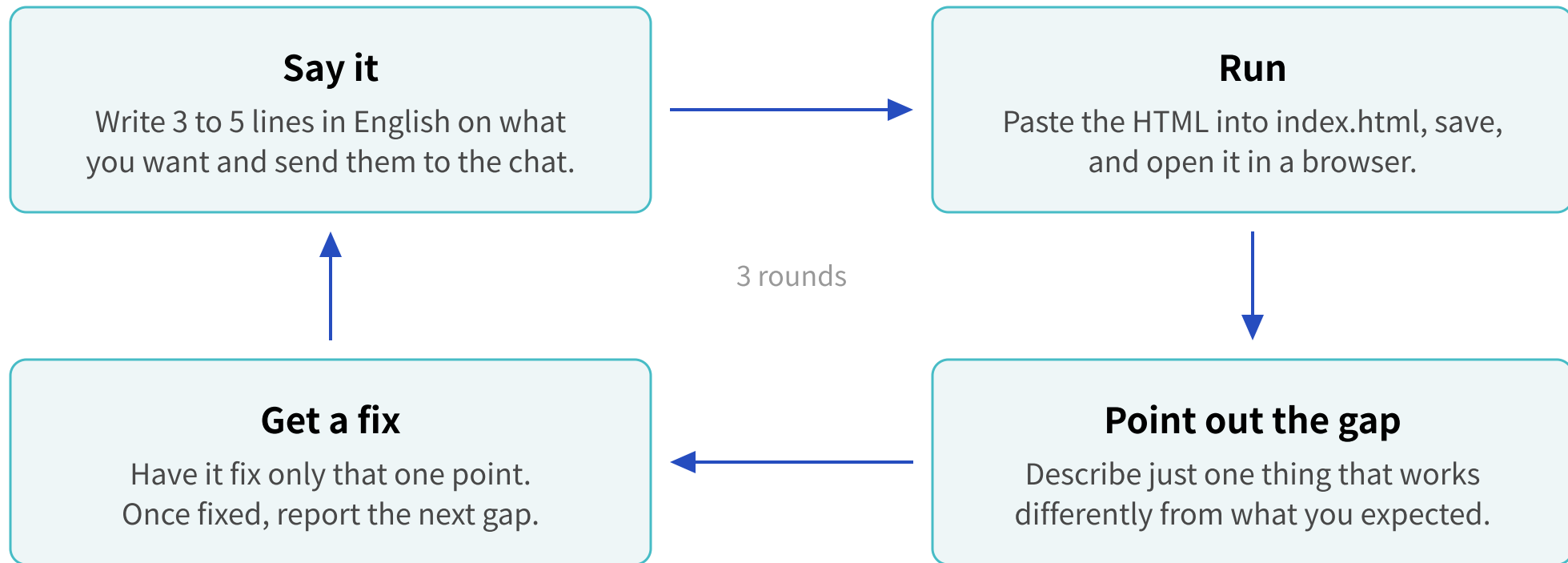
Specify, design, build, then run. You see something working only at the end. Even if you notice midway that it is not what you wanted, it is hard to go back.

Vibe coding

Loop through 3 steps: say it, run it, try it and decide. The first round shows a screen in about 5 minutes. Restate what looks wrong on the spot.

The build loop

Do not finish in one go. Shape it in 3 rounds



Round 1 passes if it works, round 2 is for the look, round 3 adds one feature. Follow this plan.

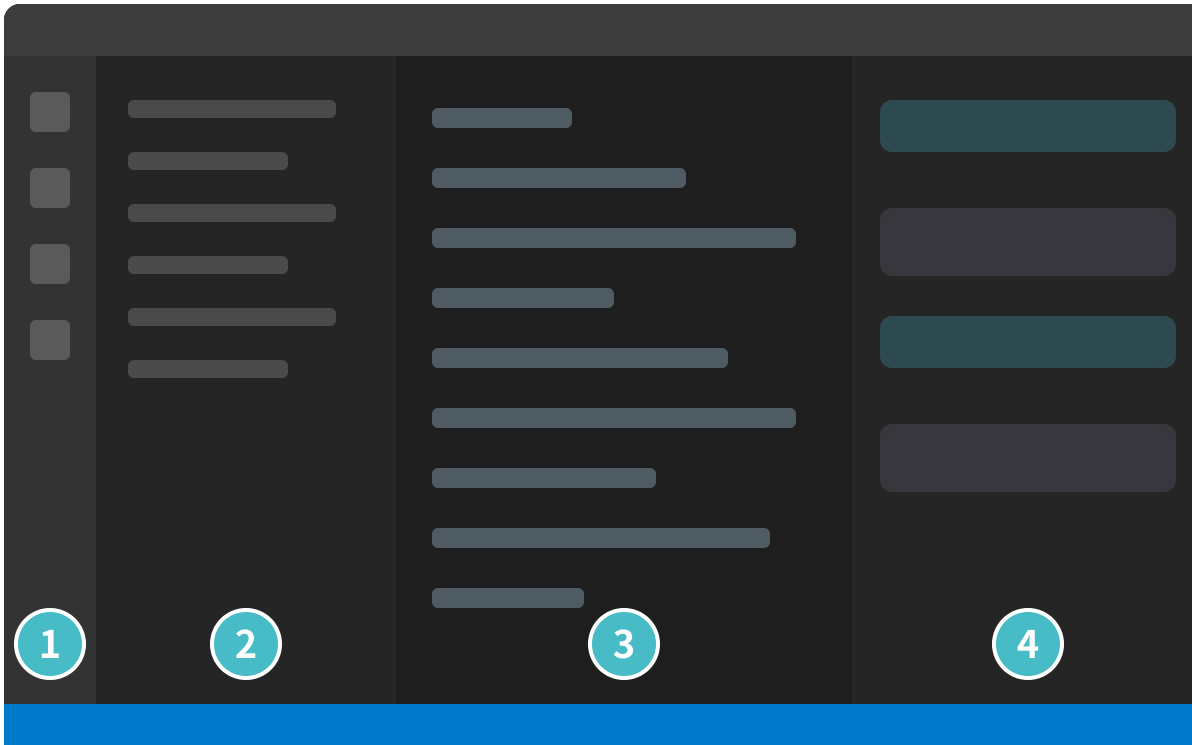
You can pick one of these 4

Aim	What to build	Sign that it works
Count things	A parts counter. Pressing the button adds 1, and Reset returns it to 0.	The number changes with each press
Convert units	mm and inch, N · m and kgf · m. Enter left, read right	The right side follows the input
No missed checks	A pre-shift inspection checklist. Shows how many are checked and how many are left	The count left drops
Game	Whack-a-mole, number guess, rock-paper-scissors. Quick results	Score up, result shown

If you have the "something to build" idea we asked for at the end of Day 1, use it. Do not use real model numbers, tolerances or client names; use made-up numbers.

Where to create files

Create new files in the Explorer



1 Activity Bar

The top icon on the far left, stacked pages, opens the Explorer

2 Explorer

Right-click under the DAY2 heading and choose New File. This is where you create index.html

3 Editor

The file you create opens here. Paste, then save with Ctrl+S

4 Chat view

The reply has a copy button at the top right. Use it instead of selecting the code by hand

Running it in the browser

Save and double-click: 2 steps to see the page

- 1 Open a folder**
In VS Code, choose File > Open Folder and open the tpro-work folder on the desktop
- 2 Create index.html**
In the Explorer, right-click > New File. Type index.html and press Enter
- 3 Paste the code**
Click copy at the top right of the Copilot reply, paste into the editor, and save with Ctrl+S
- 4 See in browser**
Open the tpro-work folder on the desktop and double-click index.html. It does not run inside VS Code
- 5 Reload after a fix**
After saving, press F5 in the browser to see your changes

If Windows hides file extensions, the file may be index.html.txt. In File Explorer, check View > File name extensions to confirm.

Write these 3 things and the first version comes back

Paste this example into the chat as is. 3 to 5 lines are enough at first. Keep the last line all day.

Create a whack-a-mole game in HTML.

- Moles pop up at random in a 3x3 grid for 30 seconds.
- Clicking a mole adds 1 point.
- Show the time left and the score at the top of the screen.
- Put everything in one HTML file and do not load any external files.

Without the last line, you may get code that loads parts from an external URL. If the company network blocks it, the page goes blank.

The goal of this exercise

Rough looks are fine. If it works, you are done

**Open it in the browser,
and something changes when you use it**

Whatever you build, this is the same finish line. Today we do not judge colors or layout

The first try usually shows a page. Use the rest of the time to shape it into something you can use.

What to say when a fix fails

Describe the symptom as is, and fixes succeed more often

Unclear

Send only "It doesn't work" or "It's buggy." The AI cannot tell what to fix, rewrites everything, and breaks parts that worked

Clear

Separate what works from what does not, like "Clicking a mole leaves the score at 0. The timer is running."

How to ask for a fix

Fix one thing at a time

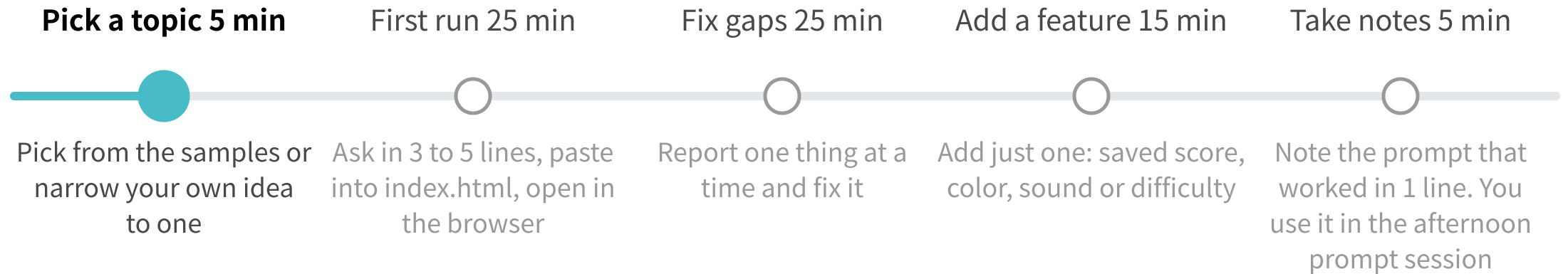
A form to paste when stuck. Write the current state, what to fix, and what to keep, in that order.

```
In the whack-a-mole game I'm building, the score stays at 0 when I click a mole.  
The countdown for the time left is working.  
Fix only the cause of the score not going up. Do not change any other behavior.  
Show the whole fixed HTML again.
```

Without "Do not change any other behavior," parts that worked may get rewritten too.

How to use the exercise time

Get it running once in the first 30 minutes



If nothing shows after 30 minutes, raise your hand. We pause the class for 3 minutes on the cause.

Exercise: Build your own project

Try it once yourself: a working app with no design document

Scenario

Listening alone will not show you how a working app appears without writing code. Once you have done it yourself, you will notice "I could just ask for this too" when you face Excel this afternoon. You also build the Day 3 mini app this way.

Steps

- 1 Pick one topic: one of the 4 samples, or the idea you thought of at the end of Day 1. Decide in 5 minutes and move on
- 2 Ask in 3 to 5 lines. Write 4 things: what to build, what to show, what happens on each action, and to keep it in one HTML file
- 3 Paste the code into index.html in the tpro-work folder, save with Ctrl+S, and double-click to open it

Exercise: Build your own project (continued)

Steps

- 1 Once it works, save a copy of the file as `index_ok.html`. If later changes break it, you can go back to this
- 2 Report one difference at a time. Asking for many at once breaks parts that worked
- 3 Finally, write the prompt that worked in `prompt.txt`, in just 1 line

Result

`index.html` in the `tpro-work` folder. `index_ok.html` and `prompt.txt` also remain. Nothing to submit.

Completion criteria

All 3 met means done. Looks do not matter

Done when

- A page appears when opened in the browser
- Clicking or another action changes something on the page
- At least one fix you asked for is in place

If you're stuck

- Blank page. First press Ctrl+S. If still blank, paste the red text from the F12 Console tab
- Code shows as plain text when you open the file. The name is index.html.txt. Turn on file extensions and rename it
- A fix does not show. You have not saved, or not pressed F5 in the browser
- It broke after more changes. Paste the contents of index_ok.html back into index.html
- The topic is too big to finish. Cut it down to one screen: fewer fields, or just one feature
- Not sure where to paste the code. Check that the Explorer shows DAY2. If not, open the folder again

Completion criteria (continued)

If time allows

- Save scores or inputs in the browser so they stay after closing the page
- Keep the layout intact at smartphone screen width
- Add a switch for difficulty or units
- Match colors to workplace notices and keep them readable when printed

Quiz

Comprehension check 1

Q. Your app opens in the browser as a blank page. What do you do first?

- A** Ask Copilot to rebuild it from scratch
- B** Paste the error text from F12 into Copilot as is
- C** Open it again in another browser
- D** Restart the PC

The answer is on the next page. Think it through yourself first.

Comprehension check 1

The answer is **B**. Pasting the error as is is fastest

Paste the error text from F12 into Copilot as is

B

The page is blank because processing stopped partway. The Console, opened with F12, shows where it stopped in red text. Paste that line as is, and Copilot fixes only the cause. You do not need to understand it.

A

Ask it to rebuild from scratch

Parts that worked get rewritten, and new bugs appear. Rebuilding is the last resort

C

Open it again in another browser

If the cause is in the code, every browser gives the same result

D

Restart the PC

A restart does not fix unsaved changes or code errors. It only costs time

No red text but still blank? You forgot to save or the file name is wrong. Check Ctrl+S and the extension.

Break

Return time goes on the whiteboard

Break 10min

The instructor writes the restart time on the board. Before leaving your seat, save index.html with Ctrl+S

The first afternoon session is prompt basics. Keep what you built open.

From practice to prompts

Keep this morning's good requests **ready to paste**

What we have done so far

This morning you wrote what you wanted in English, gave it to Copilot, and ran the returned HTML in the browser. Some requests worked on the first try, and some took 5 rewrites.



What we do next

We find where that difference came from. The first 60 minutes cover how to ask. In the 40-minute exercise, you put your usual phrases in one file, prompt.txt.



Why it matters later

This afternoon's data cleanup starts by pasting this prompt.txt. Here you also write a line that stops the AI from filling Excel gaps with made-up numbers. Day 3 analysis and reports reuse it.

03

Prompt basics

You will know the 4 parts of a request and take home one prompt you use again and again

What happens to conditions you leave out

The AI fills in what you leave out and moves on

What you left out	What the AI decided	What came back
Period	Lumped all periods together	April to June, 3 months, merged into 1 row
Grouping unit	Totals for all, not per machine	8 machines became 1 average
Output format	Chose to explain in prose	Wanted a table, got 5 lines of text
Handling blanks	Decided it could fill them	Plausible numbers filled the blank operating hours

If a drawing has no dimension, the machining side picks a reasonable value. It is the same here: whoever gets the request decides what you leave out.

The 4 parts of a prompt

Role, context, **constraints** and output format make a prompt

Role

Set who answers. Writing "as an equipment maintenance engineer at a factory" sets the level of terms and detail

Context

Write the purpose and what you have. "I have operating data for 8 machines pasted from Excel. It is for the monthly inspection report."

Constraints

Write what it must not do. "Do not invent numbers," "Do not leave units in the values," "Do not load external files"

Output format

Specify the form you want. "A 3-column table: Equipment ID / Operating hours / Defects count." A fixed form lets you compare results

Two ways to write the same request

Add the 4 parts and the form stays the same every time

Hard to fix

"Summarize the equipment data." No role, context, constraints or output format. You get 5 lines of plausible text with guessed numbers, and cannot tell what to fix. Send it again and you get yet another form.

Ready to use

With all 4 parts, the form of the reply is fixed. The full text is on the next page. We hand out this operating data in Chapter 4 this afternoon. For now, look only at the form, not the content.

Full text of the good example

Each line starts with its part. 6 lines make one prompt

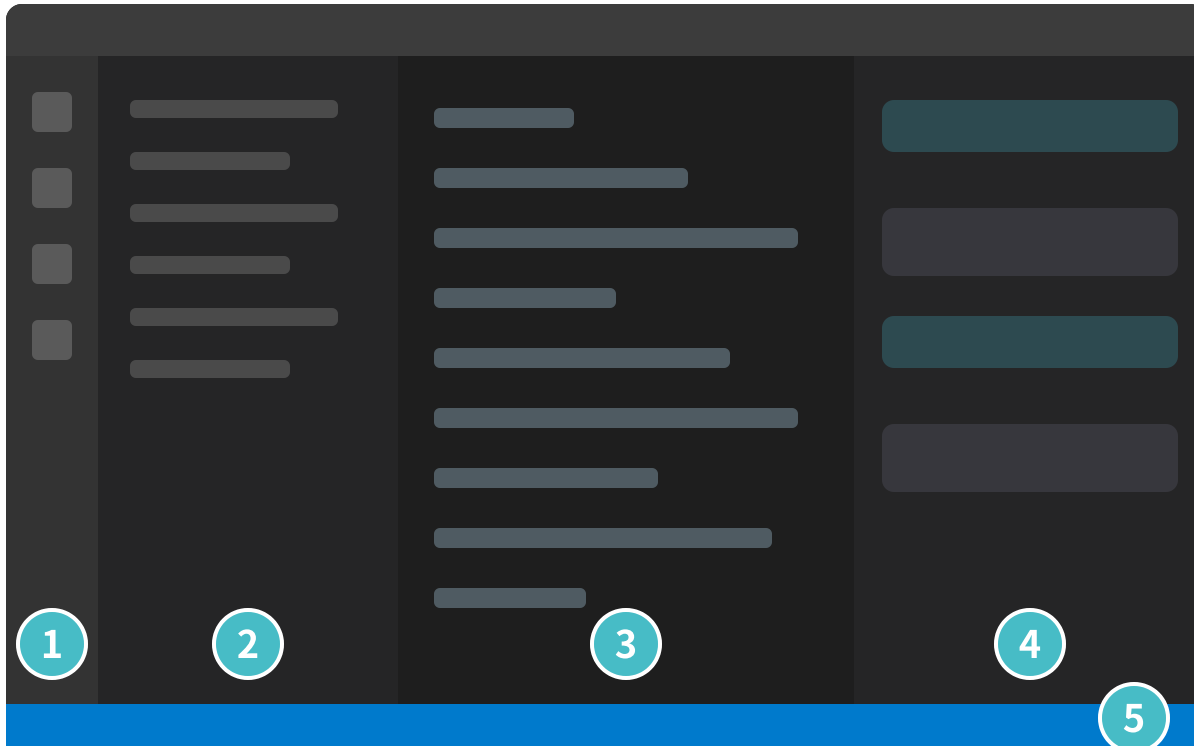
This is the full text from the right side of the previous page. The square brackets at the start of each line show which of the 4 parts it is. Do not read it aloud; just match each line to its part.

```
[Role] Answer as an equipment maintenance engineer at a manufacturing site.  
[Context] I pasted operating data for 8 machines from Excel, for a monthly inspection report.  
[Constraint] Convert operating hours to numbers. Move the unit h into the column name.  
[Constraint] Leave blanks as blanks, not 0, and give the number of blanks at the end.  
[Constraint] Do not invent values that are not in the data.  
[Output format] Return a table with 3 columns: Equipment ID / Operating hours / Defects count.
```

The brackets are only for explanation. You can remove them when sending. It also works with them.

Where to type the prompt

Type prompts in **the input box at the bottom of the right chat**



- 1 Activity Bar**
The vertical bar on the far left. The top icon, a file, opens the Explorer
- 2 Explorer**
Lists the tpro-work folder. Right-click here and choose New File to create prompt.txt
- 3 Editor**
The center. Open prompt.txt, paste the prompt, and save with Ctrl+S
- 4 Chat view**
The right side. Open it with Ctrl+Alt+I. Paste prompts in the input box at the bottom. Enter sends; Shift+Enter adds a line break
- 5 Status Bar**
The bar at the bottom. If the Copilot icon has a slash through it, you are signed out

3 lines for output format

Just adding these 3 lines after a prompt changes the result

If all 4 parts are too much, add just these 3 lines. They cover the core of output format and constraints.

- Return a table with the columns Equipment ID / Operating hours / Defects count.
- Write numbers in half-width digits only, and put units in the column names.
- Leave values that are not in the source data blank. Do not fill them by guessing.

Drop the 3rd line and plausible numbers fill the blanks. This afternoon's Excel file has 2 missing values.

How to generalize before pasting

Generalize workplace terms to industry and process level first

Client and product names

Write "a mass-production line for auto parts," not "in-vehicle connectors for Company A." Generalizing to the process and industry still gives the same cleanup steps

Model numbers and measured values

Write "Machine A" and "a value slightly over the upper limit," not "10.05mm on EQ-1420." Matching digits and units is enough for practice

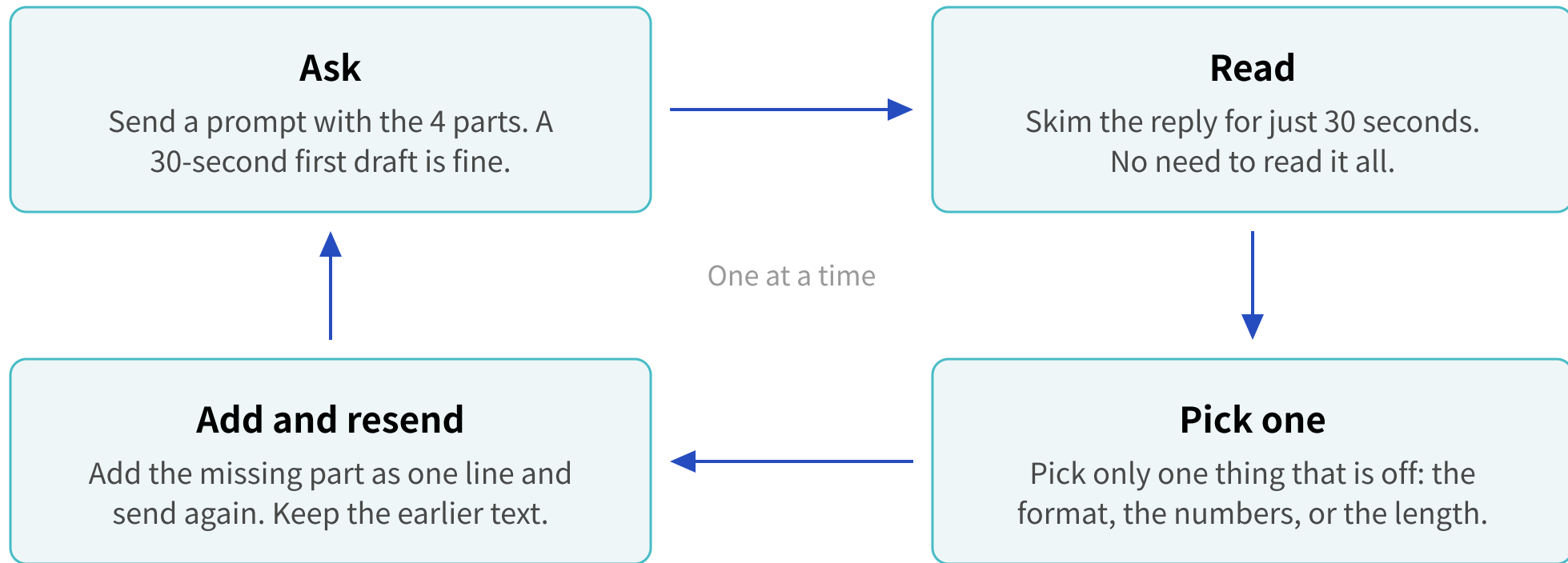
Before you click Send

Reread the text once, asking whether it is OK to share outside the company. Rereading takes 30 seconds

In a client's environment, the client's rules come first. If unsure, do not paste; take it back to check.

One round of fixes

Report **one difference at a time**. Asking for many at once changes parts that were right



How GitHub and Copilot relate

Copilot is **from GitHub**. So are sign-in and the docs



GitHub

The provider of Copilot.
The account you sign in with is a GitHub account.

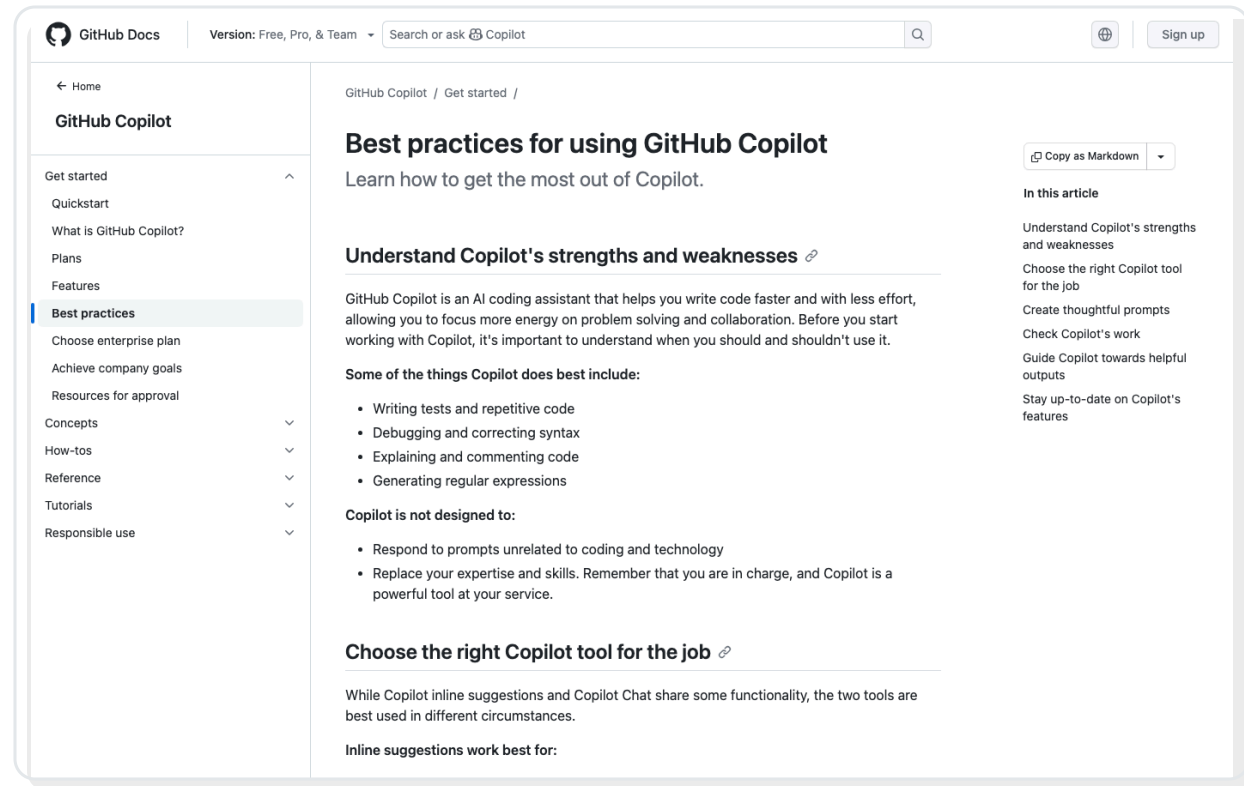


GitHub Copilot

A feature that runs inside VS Code. Its provider also publishes prompting tips.

The user name you saw at the bottom left in the morning check is the GitHub account on the left.

Be specific and always check the output. GitHub says so too



Screen: the Best practices page in GitHub's official docs

10 practice rows

The next exercise uses **these 10 rows**. Fictional practice data

Equipment	Operating hours	Stops	Defects count
EQ-001 Molding machine A	192.0	3	12
EQ-002 Molding machine B	178.5	5	18
EQ-003 Cutting machine A	205.0	2	6
EQ-004 Cutting machine B	(blank)	4	15
EQ-005 Welding robot A	210.0	1	4

3 traps: EQ-004 has blank Operating hours, EQ-006 has Operating hours as text with a unit, EQ-009 has blank Stops. This practice table is not equipment-data_2026Q1.xlsx from this afternoon.

10 practice rows (continued)

Equipment	Operating hours	Stops	Defects count
EQ-006 Welding robot B	182.5 h	6	21
EQ-007 Inspection unit A	158.0	8	27
EQ-008 Inspection unit B	166.0	7	24
EQ-009 Grinding machine A	188.0	(blank)	9
EQ-010 Paint booth A	174.0	5	16

Exercise: Rewriting the prompt

Send the same topic 2 ways and see the difference for yourself

Scenario

This morning you asked as ideas came and fixed replies as you went. For this afternoon's cleanup, that fills missing values with made-up numbers and leaves units in text. Make a rewrite pattern now, and in the afternoon you just paste it.

Steps

- 1 Copy the 10-row table from the previous page into the chat, then send only "Analyze the operating data". Skim the reply for 30 seconds (required, 5 min)
- 2 Open a new chat and paste the same table again. Add role, context, constraints and output format one by one, and resend. Always include the constraint "Do not guess values that are not in the original data" (required, 15 min)
- 3 In prompt.txt, write 3 lines on what changed from the 1st to the 2nd reply: the format, number handling and amount of explanation (required, 10 min)

Exercise: Rewriting the prompt (continued)

Steps

- 1 With spare time, remove only the "Do not guess" line and send the table with EQ-004, whose Operating hours is blank. Check that a number fills the blank.
- 2 With spare time, put the 3 lines you use most at the top of prompt.txt, so next time you can just paste them.

Result

prompt.txt in the tpro-work folder, with the full rewritten prompt and 3 lines on the differences you noticed. You use the same file for data analysis on Day 3.

Completion criteria for the prompt

Meet all 4 and you are done. Looks do not count

Done when

- The 2nd reply is a table with the columns Equipment ID, Operating hours and Defects count
- The Operating hours column has no "182.5 h" left; it holds the number 182.5
- EQ-004's Operating hours is still blank, with no made-up number
- The full rewritten prompt is pasted and saved in prompt.txt in the tpro-work folder

If you're stuck

- You get text, not a table: the output format line is buried mid-prompt. Move it to the end and resend.
- The 2nd reply looks like the 1st: you are still in the same chat. Start a new chat with the New Chat icon at the top of the Chat view, then paste.
- EQ-004 gets a number: the constraint line runs into other text. Start it with "- " to make it a separate line.
- It sends before the table is complete: Enter sends. Use Shift+Enter for a new line.
- prompt.txt is missing: it was created outside the tpro-work folder. Create it again directly under tpro-work in the Explorer.

Completion criteria for the prompt (continued)

If time allows

- Send the same request 3 ways: "Return a table", "Return CSV" and "Return a bulleted list". See which is easiest to paste into Excel.
- Change only the role to "Explain as if to a new operator" and see how the level of detail changes.
- Add one constraint line, "Sort by Operating hours, longest first", and see if it can sort too.

1st vs 2nd reply

2 parts made the difference: constraints and output format

Typical 1st reply

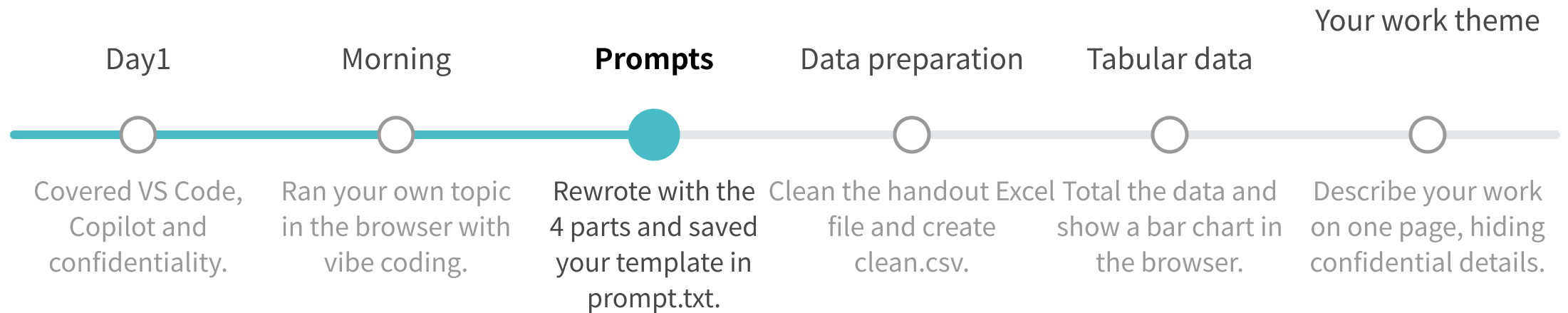
Only "Analyze the operating data" gets about 5 lines on all 10 machines. No per-machine numbers; EQ-004's blank Operating hours may get a number near the mean. The Stops total counts blanks as 0.

Typical 2nd reply

With the 4 parts, you get a 3-column table: 182.5 h becomes 182.5, and EQ-004's Operating hours stays blank. A count such as "1 blank" may follow at the end. In this form, you can compare counts with the original table.

Where we are today

prompt.txt is your starting point for the next 80 minutes



Next is a 10-minute break. Save prompt.txt before you leave your seat. The instructor will show when we resume.

From prompts to data preparation

Next, you work with **messy data like real-world Excel files**

What we have done so far

Before the break, you saved a prompt with role, context, constraints and output format in `prompt.txt`. Since this morning's vibe coding, you have used only your own topic and 10 practice rows.



What we do next

Now apply that prompt to the full data, not the practice rows. Clean the 3 sheets of `equipment-data_2026Q1.xlsx` until they match the preset check values.



Why it matters later

The next chapter totals the cleaned table and makes a bar chart. The Day 3 analysis exercises also start from a clean table. Next week you open today's `clean.csv` as is.

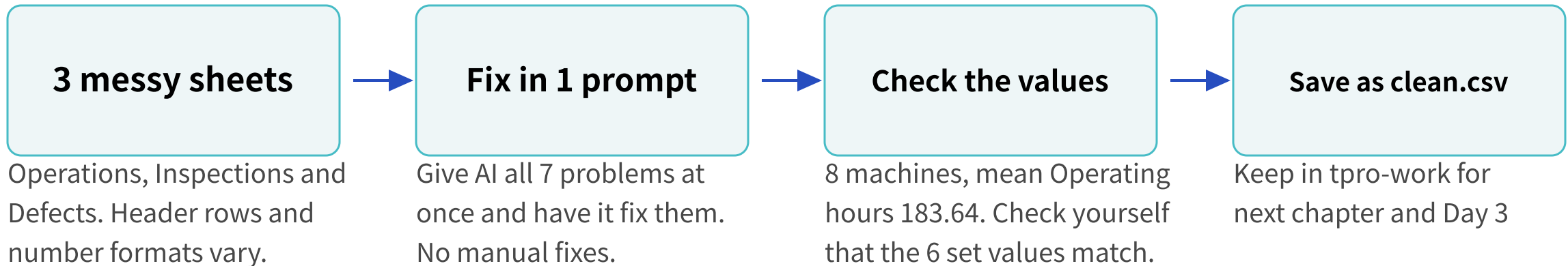
04

Data preparation

Learn to hand messy Excel data to AI and turn it into a table you can total

Goal of this chapter

By the end of this chapter, the 3 sheets become one table



Today's file

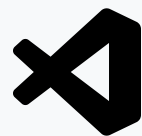
You work with all 3 sheets of equipment-data_2026Q1.xlsx

Sheet	Contents	Main problems
Operations	Operating hours, stops and defects for 8 machines	Title in row 1, headers in row 3, a blank row midway. All Operating hours values are text with units; 2 missing values
Inspections	Measured sizes and results for 10 lots	Full-width digits in values (" 1 0 . 0 5 "), dates in 3 formats, full-width hyphens in the Result column
Defects	Defect counts by process	Process is in vertically merged cells; lower rows are empty

These 3 sheets are in one file. The browser does not switch sheets for you; you pick one when you give it to AI.

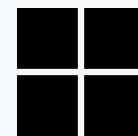
Where you write and where you run

Write in VS Code, run in the browser. **Each has its own job**



VSCode

Where you type prompts
and save the code you get
back. Nothing runs here.



Edge

Where you open and run
the saved HTML. You also
give it the Excel file here.

The screen that loads Excel is in the browser. Do not try to open Excel files in VS Code.

The browser cannot open files by itself. **You drag them in**

Open the folder

The tpro-work folder on your desktop contains 01_data-loader.html, equipment-data_2026Q1.xlsx and the lib folder.

Open the HTML

Double-click 01_data-loader.html to open it in the browser, not VS Code. A dotted drop box appears.

Drag and drop the Excel file

Drag equipment-data_2026Q1.xlsx into the dotted box. The 3 sheet names appear: Operations, Inspections and Defects.

Copy and paste

Click a sheet name, copy it as tab-separated text with "Copy this table", and paste it into Copilot Chat in VS Code. Nothing is saved yet.

Before dropping the file

If you see the dotted box, it is open in the browser



01_data-loader.html opened by double-clicking. Drop the Excel file here.

Header position and blank rows

The headers are in **row 3**, with a blank row midway

Row	How it looks in Excel	What AI receives
Line 1	FY2026 Equipment Operations (Q1)	Goes in as is, as a row with just one cell of text
Line 2	(empty)	Goes in as an empty row
Row 3	Equipment ID / Equipment name / Operating hours / Stops / Defects count	These are headers, but nothing sets them apart from rows 4 and below
From row 4	One row of data per machine	Read normally
A row midway	(empty)	AI may decide the table ends here

Vertically merged cells

Merged cells keep the value only in **the first row** of the range

Row	Process shown in Excel	Content after loading	If counted as is
Row 4	Molding (merged, 3 rows)	Molding	Counted only once
Row 5	(same cell continues)	Blank	Dropped as having no process
Row 6	(same cell continues)	Blank	Dropped as having no process
Row 7	Cutting (merged, 2 rows)	Cutting	Counted only once
Row 8	(same cell continues)	Blank	Dropped as having no process

Fill the blanks with the process name above, then count: Molding 33, Cutting 21, Inspection 7, total 61.
Without filling, Molding keeps only its first row.

Text that looks like numbers

Have AI fix units, full-width digits and dates in one go

"182.5 h" in Operating hours, " 1 0 . 0 5 " in measured values, and dates in 3 formats. They look like numbers, but you cannot calculate or sort them as is. Ask for all the fixes in one request.

Fix these 3 points.

- 1) Remove units from text like "182.5 h" and make it a half-width number
 - 2) Convert full-width digits like " 1 0 . 0 5 " to half-width
 - 3) Use the 2026-04-01 format for dates. 2026/04/01, 2026/4/2 and 2026-04-03 are mixed
- At the end, write how many cells could not be converted to numbers.

The last line prevents errors. Without it, failed cells silently become 0 and the mean drops.

Handling missing values

Filling blanks with 0 lowers the mean by **about 23 hours**

Blanks filled with 0

Count the 1 missing Operating hours value as 0 and divide by 8. The mean is about 160.7 hours. It reads as if that machine was stopped.

Blanks excluded

Exclude the 1 missing value and divide by 7: the mean is 183.64 hours. Always note how many were excluded, to show it is a mean of 7 machines.

Prompt for cleaning

In the exercise, **copy these 7 items as is**

This lists the problems from the last 5 pages as 7 items. Do not write your own; copy this text, then paste the table.

As an equipment maintenance engineer, clean up the table pasted below.

- 1) The headers are in row 3. Ignore rows 1 and 2 and any blank rows in between
- 2) If the Process column is empty, fill it with the process name from the row above
- 3) Remove units from values like "182.5 h" and make them half-width numbers
- 4) Convert full-width digits like "1 0 . 0 5" to half-width
- 5) Use the 2026-04-01 format for all dates
- 6) Do not fill blanks with 0. Leave them blank and count them
- 7) At the end, write the number of cells that could not be converted to numbers

Checking the cleanup

If these 6 values match, your data preparation worked

Check item	Expected value
Number of machines	8
Missing Operating hours	1
Missing Stops	1
Mean Operating hours	183.64 hours
Median Operating hours	182.5 hours
Cells that could not be converted to numbers	0

Exercise: Running the data preparation

Replace your monthly manual fixes with **a single prompt**

Scenario

Many of you fix the monthly equipment Excel by hand before totaling: move headers, delete units, fix full-width digits. See if today's single prompt can replace that work.

Steps

- 1 In the handout folder, double-click 01_data-loader.html to open it in the browser, and drag equipment-data_2026Q1.xlsx into the dotted box. Click the Operations sheet and copy it with "Copy this table" [5min]
- 2 Open Copilot Chat and paste the 7 items from the "Prompt for cleaning" page first. Right below them, paste the copied Operations sheet and send [10min]
- 3 Check the returned table against the 6 check items one by one: 8 machines / 1 and 1 missing / mean 183.64 / median 182.5 / 0 unconverted cells [8min]

Exercise: Running the data preparation (continued)

Steps

- 1 Tell the chat only the items that did not match, and have them fixed. Redoing everything breaks other parts [4min]
- 2 Copy the cleaned table. In the left pane of VS Code, right-click, choose New File, name it clean.csv, paste, and press Ctrl+S [5min]

Result

clean.csv in the tpro-work folder. It is done when it shows 1 header row and 8 machine rows in VS Code.

Completion criteria for data preparation

When all 3 are met, you can move on to the next chapter

Done when

- clean.csv is in the tpro-work folder and shows 8 machine rows in VS Code.
- At least 5 of the 6 check items match. 8 machines and mean 183.64 hours must both match.
- The Operating hours column has no "h" and no full-width digits left.

If you're stuck

- The machine count is 9 or 10. It counted the title row or a blank row. Resend only item 1 of the prompt.
- The mean is in the 160s. Missing value filled with 0. Resend only item 6 of the prompt.
- Unconverted cells are not 0. Values with units or full-width digits remain. Follow up with "List the cells that were not converted."
- The reply is cut off. You pasted all 3 sheets. Paste only the Operations sheet again.
- clean.csv will not save. In File Explorer, turn on View > File name extensions and check that it is not clean.csv.txt.

Completion criteria for data preparation (continued)

If time allows

- Write in 1 line how Operating hours relate to Stops. You need it in the next chapter.
- Apply the same prompt to the Inspections sheet and find which of the 7 items you must change to make it work.

Before adding, **make the units match**

Written as	In minutes	If added as is
90分	90分	Becomes 90
1.5h	90分	Becomes 1.5
1h 30min	90分	Cannot be added. Treated as text

maintenance-records_2026H1.csv mixes these 3 forms. Fixing 96 rows by hand is not realistic.

Exercise: Totaling maintenance records

Unify units and names **first**, then total by assignee

Scenario

You report maintenance time by assignee every month. Several people write the records, and times and names are written in different ways.

Steps

- 1 Drop maintenance-records_2026H1.csv onto 01_data-loader.html and view it.
- 2 Paste the table into Copilot Chat and ask for 3 things at once: all durations in minutes, one spelling per assignee, no blank or duplicate rows
- 3 From the cleaned table, have it total the time per assignee.
- 4 Check that the grand total is 10938 min.

Result

A table of total time for the 4 assignees. Grand total: 10938 min (182.3 hours).

Whether names were merged **shows in the head count**

Done when

- Assignees are merged into 4. Without unifying names, they split into 8 or 10.
- There are 96 valid rows, after removing 1 blank row and 1 duplicate.
- The grand total is 10938 min, or 182.3 hours.

If you're stuck

- The head count is wrong. The name instruction did not work. Say clearly that ヤマダ, Taro Yamada and Yamada are the same person.
- The total is too large. A duplicate row remains. Say there is 1 row that is exactly the same.
- The total is too small. Rows written like 1h 30min were dropped. Check that they are converted to minutes too.

If time allows

- Count records per Category. You get Cleaning 24, Adjustment 21, Repair 20, Parts replacement 18, Periodic inspection 13.
- Total maintenance time per machine and name the machine that needs the most work.

Q. What happens if you load vertically merged process names as is and total them?

- A** Every row gets a process name, just as it looks in Excel
- B** Merged cells cannot be loaded, so those rows disappear
- C** Only the first row counts with a process name; the rest have none
- D** Blank process names are treated as the number 0

The answer is on the next page. Think it through yourself first.

Comprehension check 2

The answer is **C**. Fill the blanks first, and Molding counts 33

Only the first row counts with a process name; the rest have none

C

Merged cells only look grouped on screen; only the first cell of the range holds a value. After loading, the other rows are empty, so counting by process leaves only the first row of Molding. Fill the blanks with the process name above, then total: Molding 33, Cutting 21, Inspection 7, 61 in total.

A

Every row gets a process name

That is how it looks on screen, not what is in the cell. Click it and the formula bar is empty

B

Those rows disappear

The rows remain. Only the Process column is empty; Defect type and Count come in as is

D

Blanks are treated as the number 0

Blanks come in empty, not as 0. Making them 0 is a choice made when totaling, not in the source data

You will use 33 and the total of 61 to cross-check the chart exercise in the next chapter.

From data preparation to totals

Turn the cleaned table into **readable numbers and one chart**

What we have done so far

You cleaned the Operations sheet and saved it as clean.csv: 8 machines, mean 183.64 hours.



What we do next

Next, you use the other 2 sheets. Pull out-of-spec lots from Inspections, total Defects by process, and make a bar chart. It all runs in the browser.



Why it matters later

You will reuse charts and cross-checks in the Day 3 report. Today, take home a pattern for checking AI's numbers yourself.

05

Working with tabular data

Learn to total cleaned data and show a chart in the browser

Operating hours statistics

Look only at the mean and you miss the **spread**

Item	Value	How to read it
Mean	183.64 hours	Mean of 7 machines, excluding the 1 missing value
Median	182.5 hours	Close to the mean. No machine is far off
Standard deviation	18.51 or 17.13 hours	Depends on the formula: 18.51 for a sample, 17.13 for a population. Either way, read it as about ± 18 hours
Minimum	158.0 hours	One machine runs about 25 hours less than the others
Maximum	210.0 hours	52 hours more than the minimum

These Operating hours are per-machine figures from the Operations sheet. The defect counts by process that come next are totaled on a different basis.

Exercise: Extracting out-of-spec lots

Use a condition to pull out-of-spec lots from 10 lots

Scenario

If you scan measured sizes by eye, you misread digits. The values also include full-width digits. Write a condition and let the machine find what people miss.

Steps

- 1 Drag equipment-data_2026Q1.xlsx onto 01_data-loader.html, click the Inspections sheet, copy it with "Copy this table" and paste it into Copilot Chat [2min]
- 2 Ask: "Convert full-width digits to half-width and format dates as 2026-04-01" [2min]
- 3 Then ask: "Measured size spec: lower limit 9.9, upper limit 10.1. List the lot IDs out of spec." [2min]
- 4 Note the lot IDs you get. Also count how many OK and NG are in the Result column of the source data [2min]

Result

No deliverable. You are done when 3 out-of-spec lot IDs are listed on screen.

Completion criteria for out-of-spec extraction

Correct if you get 3 and the total comes to 10 lots

Done when

- 3 out-of-spec lot IDs are listed: L-1003, L-1006 and L-1009.
- The source Result column has 6 OK and 3 NG. The 3 lots from AI have the same lot IDs as the 3 NG.
- 6 OK + 3 NG + 1 counted as neither = 10 lots in total.

If you're stuck

- You get 4, and 1 has a blank value. It counts the blank lot as out of spec. Say again that lots with no value cannot be judged.
- You get none, or the count changes each time. It compares with full-width digits left. Redo the half-width step first.
- More than 3. Writing the upper limit as "10.1 or more" counts exactly 10.1 as out of spec. Change it to "above the upper limit 10.1".
- It does not match the Result column. Some rows have a full-width hyphen in Result. Treat those rows like blanks.

Completion criteria for out-of-spec extraction (continued)

If time allows

- Also check that no lot falls below 9.9. It is easy to check only the upper limit.
- Narrow the spec to 9.95 to 10.05, see how many you get, and confirm that the threshold changes the conclusion.

Checking the extraction results

The instructor demos this. You start with the chart exercise

- 1 Give the spec as numbers**
Pass numbers: "Measured size spec: lower limit 9.9, upper limit 10.1." "Spec" alone is not enough.
- 2 Fix full-width digits before judging**
Convert " 1 0 . 0 5 " to half-width before comparing. Compared as text, values sort in the wrong order.
- 3 List the lots out of spec**
3 lots, L-1003, L-1006 and L-1009, exceed the upper limit 10.1.
- 4 Compare with the original Result column**
The source Result column has 6 OK and 3 NG. Check that the lot IDs match the 3 from AI.
- 5 If they differ, check the source**
1 row has a blank value and a full-width hyphen in Result. Count this row as neither.

6 OK, 3 NG and 1 missing make 10 lots. If the total is not 10, you have missed a count somewhere.

Reading correlation

A correlation of -0.965 **does not prove a cause**

What -0.965 shows

The numbers show a trend: machines with longer Operating hours have fewer defects. The closer to 1, the straighter the line.

What it does not show

You cannot say running longer reduces defects. The busy machines may just happen to be newer.

Check the sample size first

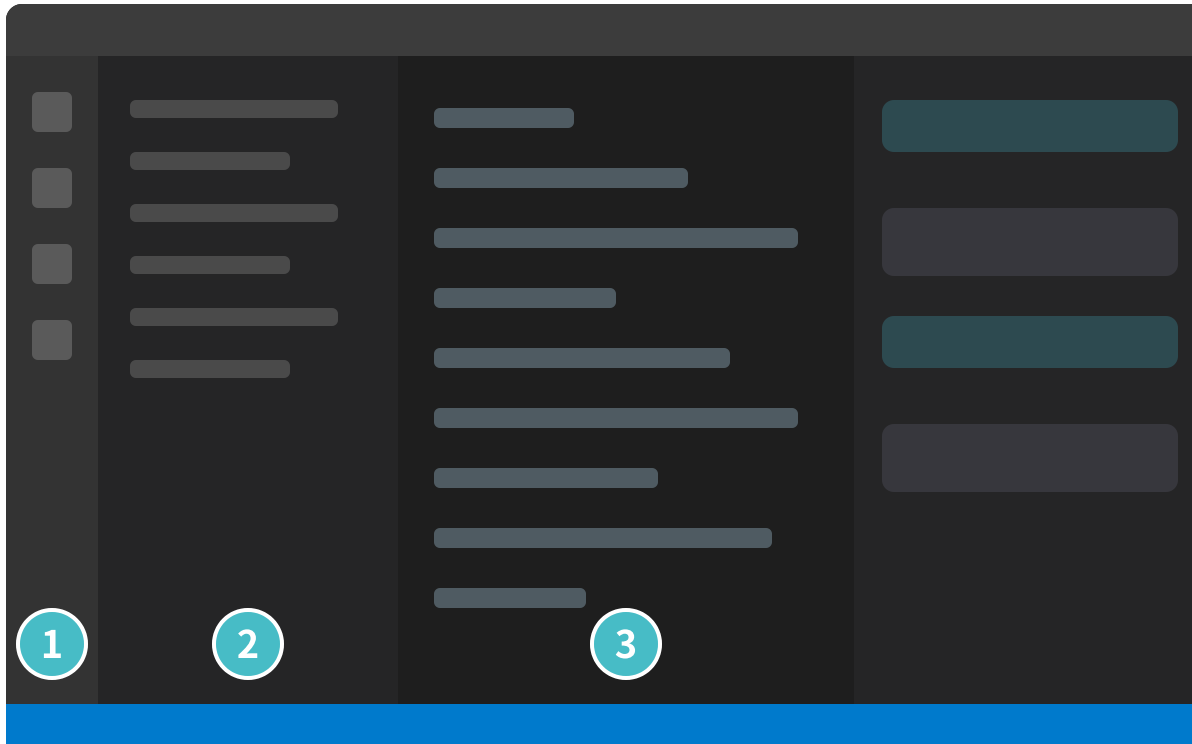
It is from 7 machines; the one with blank Operating hours is excluded. Dropping 1 machine moves the value. Always state the count.

What to check next

Add other columns, such as install year or maintenance frequency, to check. Use correlation to decide where to look next.

Preparing for the chart exercise

Copy the chart library into tpro-work before using it



1 Activity Bar

The leftmost vertical bar. Click the top file icon to open the Explorer.

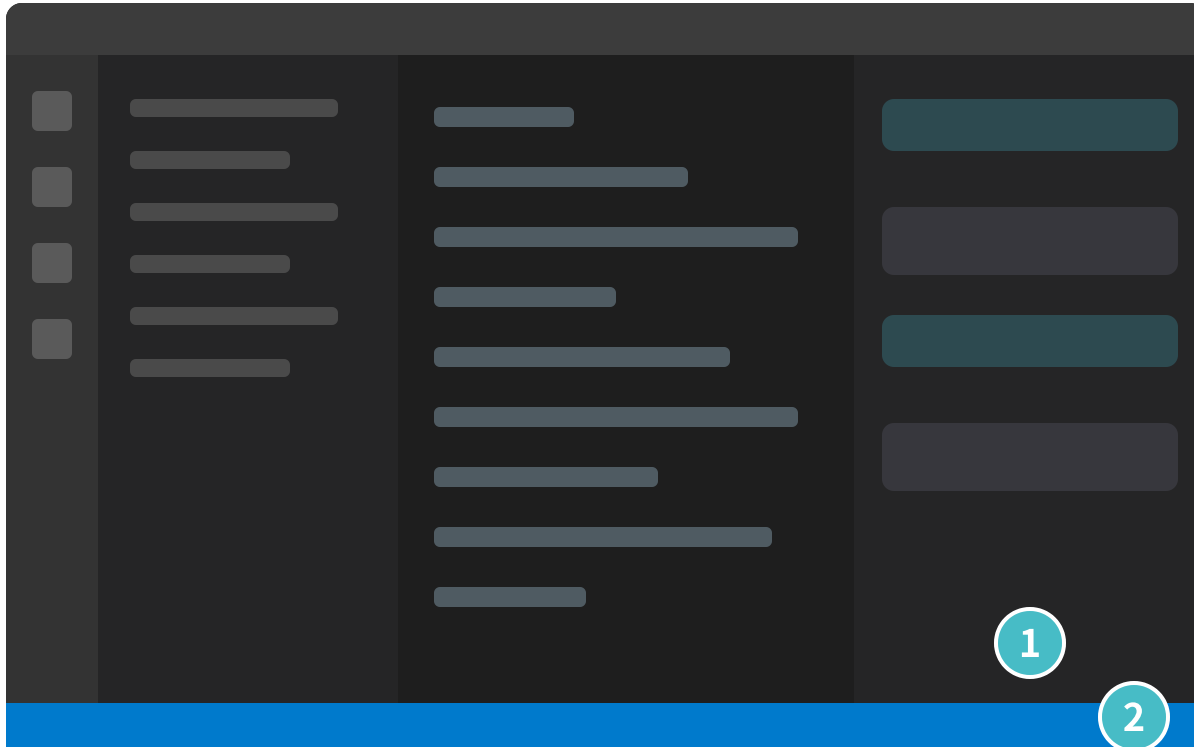
2 Explorer

Shows the tpro-work folder. If chart.umd.js is listed here, the copy worked.

3 Editor

Where chart.html opens. A dot next to the tab name means unsaved. Press Ctrl+S and the dot goes away.

Preparing for the chart exercise (continued)



1 Chat view

The right side. Paste the prompt here. Copy the code you get with the copy button at the top right of the code box.

2 Status Bar

The bottom bar. Check that the encoding is UTF-8. If text is garbled in the browser, fix it here.

Prompt for the chart

Name the file to load, or the chart stays blank

Send these 5 lines right after the pasted totals. Without line 3, AI tries to load the library from an internet URL, and the chart does not show in your company environment.

```
Create one HTML file that shows these totals as a bar chart.  
Load chart.umd.js from the same folder with <script src="chart.umd.js"></script>.  
Do not load anything from URLs on the internet.  
X-axis: process name. Y-axis: defect count. Show the count above each bar.  
Return only the HTML. No explanation needed.
```

Add this prompt to prompt.txt too. On Day 3, change only the axis line to make other charts.

Turn your totals into a chart you can show at a meeting

Scenario

Reading out defect counts by process does not show which process is worst. Today, make one page you can show at a meeting: one HTML file that opens in the browser.

Steps

- 1 Open the lib folder in the handout folder and copy chart.umd.js to tpro-work: right-click > Copy, then right-click in tpro-work > Paste [3min]
- 2 Show the Defects sheet in 01_data-loader.html and copy it with "Copy this table". Ask: "If Process is empty, fill it with the process name from the row above" [5min]
- 3 Then ask: "Count the defects for each process." It is correct if you get Molding 33, Cutting 21, Inspection 7 [5min]

If you ask for Step 3 without Step 2, the counts will not be 33, 21 and 7. Fill the merged cells first.

Exercise: Totals and a chart (continued)

Steps

- 1 Paste the 5-line prompt from the previous page to create the HTML. Copy the returned code and save it as chart.html in the tpro-work folder [6min]
- 2 Double-click chart.html to open it in the browser. Check that bar heights and the numbers on the bars match your totals [5min]

Result

chart.html in the tpro-work folder. Double-click it and the browser shows 3 bars.

Completion criteria for the chart exercise

Done when 3 bars show with 33, 21 and 7

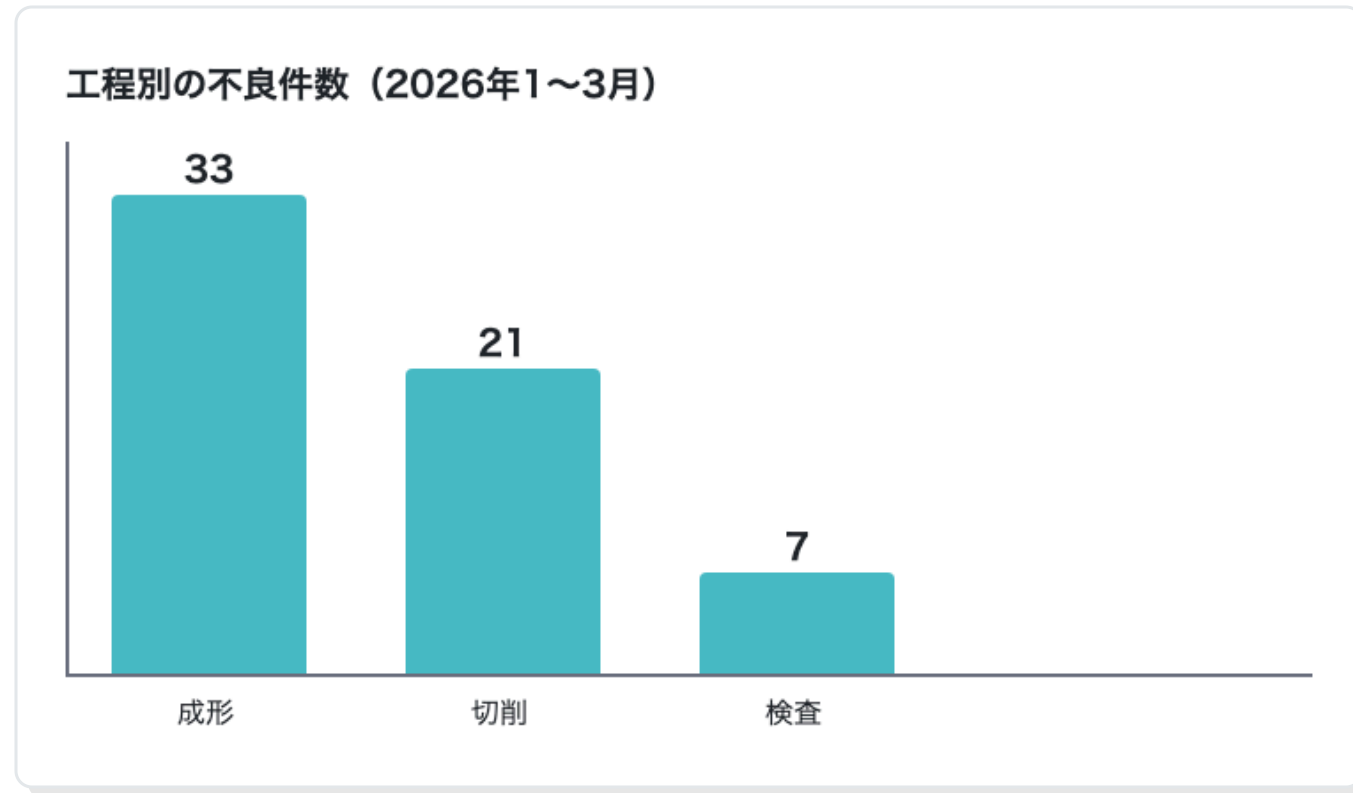
Done when

- chart.html is in tpro-work, and double-clicking it shows 3 bars in the browser.
- Bar heights go Molding > Cutting > Inspection.
- The numbers on the bars are 33, 21 and 7, which add up to 61.

If you're stuck

- The browser is blank. chart.umd.js is missing from tpro-work, or the HTML tries to load a URL starting with http.
- Only 1 bar shows. Blank process cells were not filled, so only Molding was counted. Redo the fill request.
- The total is not 61. Some rows were counted while still blank. Show it the filled table again and have it recount.
- Double-clicking opens VS Code. Right-click, choose Open with, and pick the browser.
- Text is garbled. Set the encoding in the Status Bar to UTF-8 and save again.

3 bars with 33, 21 and 7. That is the finished chart



A sample of chart.html opened in the browser. Bar colors and widths may differ.

Completion criteria for the chart exercise (continued)

If time allows

- Make a scatter plot of Operating hours vs. Defects count and see what -0.965 looks like. The machine with blank hours cannot be plotted, so there are 7 points.
- At the end of chart.html, add a 1-line comment: which works better in your meetings, comparison by machine or breakdown by process?

New data shows whether you have learned the steps

Done when

- Open equipment-data_2026Q2.xlsx and clean Operations with today's steps. It has 12 machines and 1 missing Operating hours value.
- If the mean Operating hours after cleanup is 185.86 hours, the data preparation worked.
- Out-of-spec rows in Inspections come to 5 (L-2003, L-2006, L-2009, L-2013, L-2018).

If you're stuck

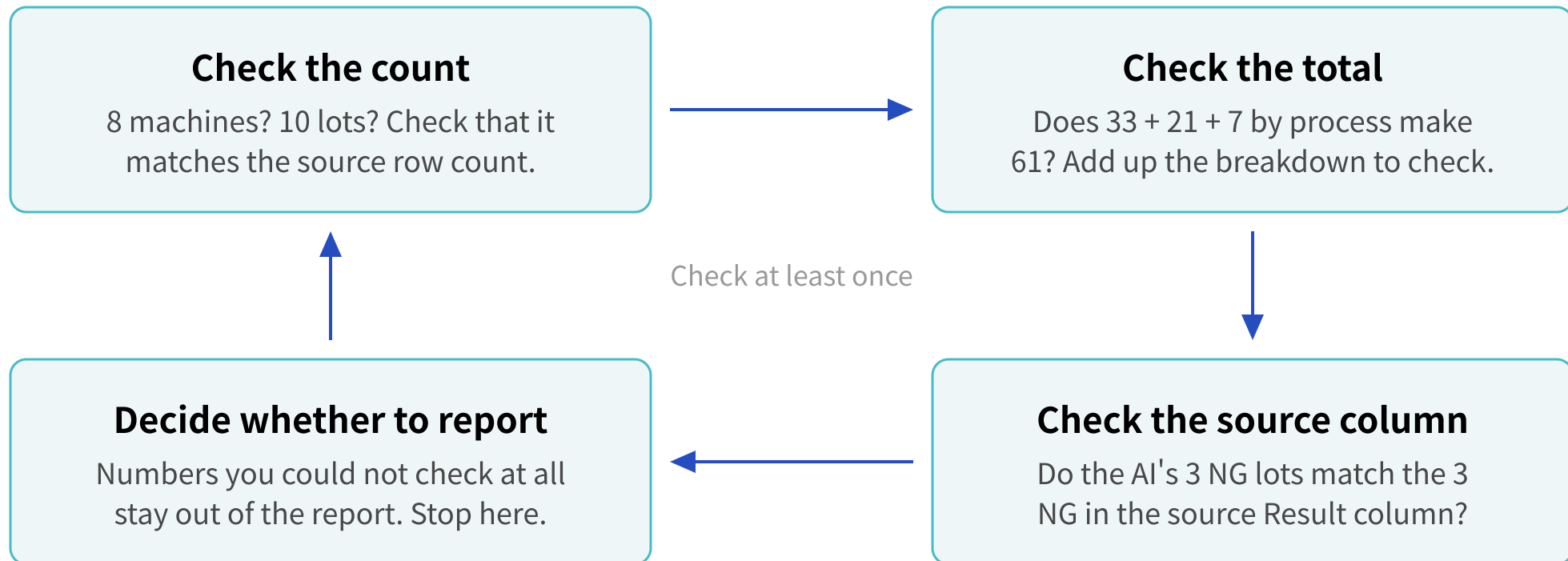
- Only 11 machines. The row written in full-width digits, 2 1 0.5 h, was not converted.
- The mean does not match. The unit is written 2 ways, h and hours. Convert both to numbers.
- Only 4 out of spec. One value written in full-width digits, like 9 . 8 5 , is missing.

If time allows

- Total Defects by Process: Molding 46, Cutting 32, Press 15, Inspection 8, 101 in all.
- Result in Inspections is written 3 ways: OK, ○ and Pass. Unify them, then count again.
- Compare Q1 and Q2 and write in 1 line how the mean Operating hours changed.

Cross-check pattern

Check every number from AI against something **at least once**



Today you used all 3 checks. On Day 3 you use them in the same order for the report.

From handout data to your own work

Leave the handout data and write about **your own workplace**

What we have done so far

So far, everyone used the same fictional Excel file. Cleanup, extraction and charts all had fixed answers.



What we do next

In the last 30 minutes, write down one chore from your own workplace. No submitting or presenting. You keep what you write.



Why it matters later

Your theme becomes the topic of the mini app at the end of Day 3. Without one, Day 3 afternoon starts with a hunt for a topic.

06

Your work theme

You can describe your own work problem in a way that is safe to share outside the company.

4 rules for writing

No submitting or presenting. Write it for your own notes

Rule	In practice
Generalize names to industry and phase	Write "design process in the auto industry", not a specific company name or a machine model number.
No real data, real files or screenshots	Do not paste screens of your own Excel files or drawings. Replace them with made-up values like today's exercise data.
Write goals and issues, not exact figures	Write "judging dimensions takes time", not actual tolerance values.
Check yourself that it is safe to share	Read it again when you finish. Leave out any part you are unsure about.

These 4 are TechnoPro's rules as they are. Use the left column without rewording it.

Sample write-up

Fill in these **5 lines** and you have your theme

Not OK: "I re-paste CSV from our internal system into Company B's controller test reports. 40 min each."
It names the client, product and system path. Rewrite it in the 5 lines below.

```
[Industry and process] Inspection process in the electrical industry  
[Task name] Create reports from a list of test results  
[Current way] Copy from the list into the report template by hand  
[How much effort] Tens of minutes per report, dozens of reports a month  
[Part for AI] Copying and formatting
```

The problem and the hope remain; client names, product names and exact numbers are gone. Describe effort only as "tens of minutes" or "dozens".

Exercise: Write down your theme

Fill in 5 lines in 12 minutes. We do not collect them

Scenario

On Day 3 afternoon you have 130 minutes to build a mini app for your work. Choosing a topic then leaves no time to build. Pick one chore from your workplace today.

Steps

- 1 On the left side of VS Code, right-click > New File, type theme.txt [1min]
- 2 Copy the 5 headings from the previous page and fill in the right side in your own words. If you get stuck, start with this week's most tedious task [7min]
- 3 Narrow the part for AI to one thing. Pick a part, such as only copying or only totaling, not the whole task [2min]
- 4 Reread what you wrote, and find and delete any client names, product names or exact numbers [2min]

Result

theme.txt in the tpro-work folder, with all 5 lines filled. No collecting or presenting.

Completion criteria for the theme write-up

Done when each of the 5 has one sentence

Done when

- Industry and process, Task name, Current way, How much effort and Part for AI each have one sentence.
- Effort is given roughly, like "tens of minutes" or "dozens a month", with no exact numbers.
- The part for AI is narrowed to one thing. It points to part of the task, not the whole task.
- It is saved as theme.txt in the tpro-work folder.

If you're stuck

- The theme is too big. Narrow it to one task you did this week, not your team's whole job.
- You cannot think of a task. Think about where in your work today's totals or charts could fit.
- Without proper names, it makes no sense. Fill in the industry and the process first, and the rest gets easier.
- Unsure whether you may write it. If unsure, do not write it. Leaving it blank is fine.

Completion criteria for the theme write-up (continued)

If time allows

- Write the task in 3 steps, the way you do it now. On Day 3 you use this to decide which part to give to AI.
- If people in another process do the same task, add one line on what differs.

What you made today

Check that the tpro-work folder has all 5 files

index.html

A browser page from the morning vibe coding exercise.

prompt.txt

Template text from the early afternoon prompt exercise, plus prompts for cleanup and charts.

clean.csv

The Operations table from the data preparation exercise. Used on Day 3.

chart.html

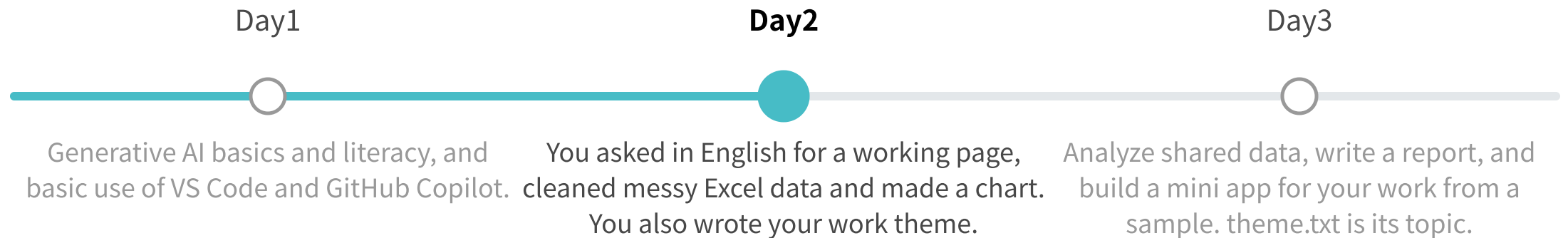
The bar chart page from the chart exercise. It runs with chart.umd.js in the same folder.

theme.txt

Your work theme. It becomes the topic of the Day 3 mini app.

What you do on Day 3

Next week you continue in the same setup and folder



There is only one thing to do before next time. Reread theme.txt and add to it if you want to change anything.



Givery, Inc.

Teito Shibuya Bldg. 8F, 15-13 Nampeidai-cho, Shibuya-ku, Tokyo 150-0036, Japan