The AI Bootcamp

17

Turn Today's Prompt Into a Template

In this chapter you'll learn

The reusable shape hiding inside a one-off prompt, a 30-second capture habit, and a library you will still open in month 3

The AI Bootcamp

THE PROBLEM

Most AI education is a pile of disconnected tips. A LinkedIn carousel here, a YouTube tutorial there, a $2,000 cohort course that goes out of date before the refund window closes. None of it is built to be read in order, so nothing compounds. You finish twelve videos and can recite twelve facts, not perform one new skill.

THE FIX

The AI Bootcamp is built the other way. 90 skills, packed 2 to 3 per chapter, 36 chapters, 3 tracks. Every chapter states what it needs from you and what it hands to the chapter after it, so skill 47 only works because skill 12 already landed. Read it in order and by the end you are not collecting facts about AI, you are running work through it.

LETTER FROM THE EDITOR

We built this because we kept meeting smart people who felt behind on AI for no reason other than nobody had organized the material for them.

This book does not assume you are technical, and it does not water anything down either. Every chapter earns its place: a real skill, a real exercise, a one-page companion you can flip back to in month three when you have forgotten the details but still need the answer.

If a chapter loses you, or a chapter changes how you work, I want to hear about it either way.

Kris

Editor, AI Central

Write to me at **editor@thecentral.ai**

THE 3 TRACKS

AI 101

Get running

Pick a tool, set it up, write a clear instruction, use it safely at work.

AI 202

Prompt properly

The framework behind every good instruction, and catching AI when it is wrong.

AI 303

Build without watching it

Connect AI to your tools, and automations that run while you are not looking.

IN THIS LESSON YOU'LL LEARN

By the end of this chapter you can take a prompt that worked, separate the part that repeats from the part that was about today, mark the swappable pieces so anyone can refill them, capture it in the half-minute after it lands, and store it somewhere you will find it again without searching a chat history.

You have written genuinely good prompts across the last four chapters. Almost all of them are gone. They are sitting in a conversation you will never scroll back to, which means every week you rebuild work you already did once. This chapter is about keeping it.

CONTENTS

17.1	A prompt that worked is an asset, and you keep throwing it away
17.2	Every prompt that worked has a part that changes and a part that never does
17.3	The moment to save a prompt is the moment it works
17.4	Keep them where you will actually look

TOPICS REQUIRED

Chapters 13 to 16. You need the framework from Chapter 13, and you need at least one prompt from the last four chapters that actually produced something you used. Go and find it before you read on. If you cannot find it, that is the chapter making its own point.

AFTER THIS CHAPTER

Chapter 18 assumes you have a template or two written down, and somewhere to put the next one. It hands you a job worth templating: reading a long document.

17.1 A prompt that worked is an asset, and you keep throwing it away

Ask yourself where your best prompt from last week is right now. The honest answer is somewhere in a conversation you are never going to scroll back through.

Standing instruction

Set once, then applies by itself to every conversation. Chapter 11 built these.

TRAITS

· Works whether or not you remember it exists

· One set, shared across every task you do

· Best for who you are: your role, your tone, your defaults

PROS

+ Costs nothing at the moment you need it

+ Fixes the same recurring annoyance everywhere at once

CONS

– Cannot get task-specific without affecting unrelated work

– Easy to forget it is switched on, and then blame the model

Template

Written down deliberately, and does nothing at all until you reach for it.

TRAITS

· Sits still until you go and get it

· One per job, and you can keep as many as you like

· Best for what you do: a task you repeat with different inputs

PROS

+ Carries detail far too specific to leave running all the time

+ Dozens can coexist without ever conflicting

CONS

– Worthless if you cannot find it inside 20 seconds

– Somebody has to write it down, and that somebody is you

Fig 17.1 Two ways to stop repeating yourself, and they are not interchangeable. The test is whether you want it running when you are not thinking about it, or waiting quietly until you go and get it.

17.1

Chapter 11 owns one half. Standing instructions and memory apply automatically to everything you type.

This chapter owns the other. Templates stay dormant until you invoke them, which is exactly why they can be specific.

The test. Would you want this running on every conversation you have? If not, it is a template.

Here is the problem this chapter exists to fix. Over the last four chapters you have written prompts that worked, some of them after three or four rounds of fixing. Every one of them is now buried in a thread, below a dozen follow-ups, in a history you search by keyword and mostly by luck. So next month you write the same prompt again, slightly worse, from memory.

The waste is not the typing. It is the thinking. A prompt that took you four attempts to get right contains four corrections you will not remember making, and rebuilding it from scratch means paying for those corrections a second time.

Chapter 11 already solved one version of this problem, and it is worth being precise about which version, because the two get muddled constantly. Standing instructions, memory and custom setups are things you configure once and then forget, and they apply to every conversation automatically. That is their whole value: they work when you are not paying attention.

A template is the opposite kind of object. It sits in a file doing nothing until you deliberately go and get it. That sounds like a weakness and it is actually the point, because something that only fires when you ask for it can be as narrow and opinionated as you like. You would never set "always give me six bullets, then three questions to ask before the next call" as a standing instruction. It would ruin every casual question you asked for the rest of the year. As a template you reach for on vendor proposals, it is exactly right.

So the test is short. Would you want this behaviour on for every conversation you have? If yes, it belongs in Chapter 11 and you should go back and set it. If no, and it is still something you do more than once, it belongs here.

Go and find one prompt from the last four chapters that produced something you actually used. That is the raw material for the rest of this chapter.

17.2 Every prompt that worked has a part that changes and a part that never does

The skill is not writing a template from nothing. It is looking at a prompt you already wrote and seeing which half of it was only about today.

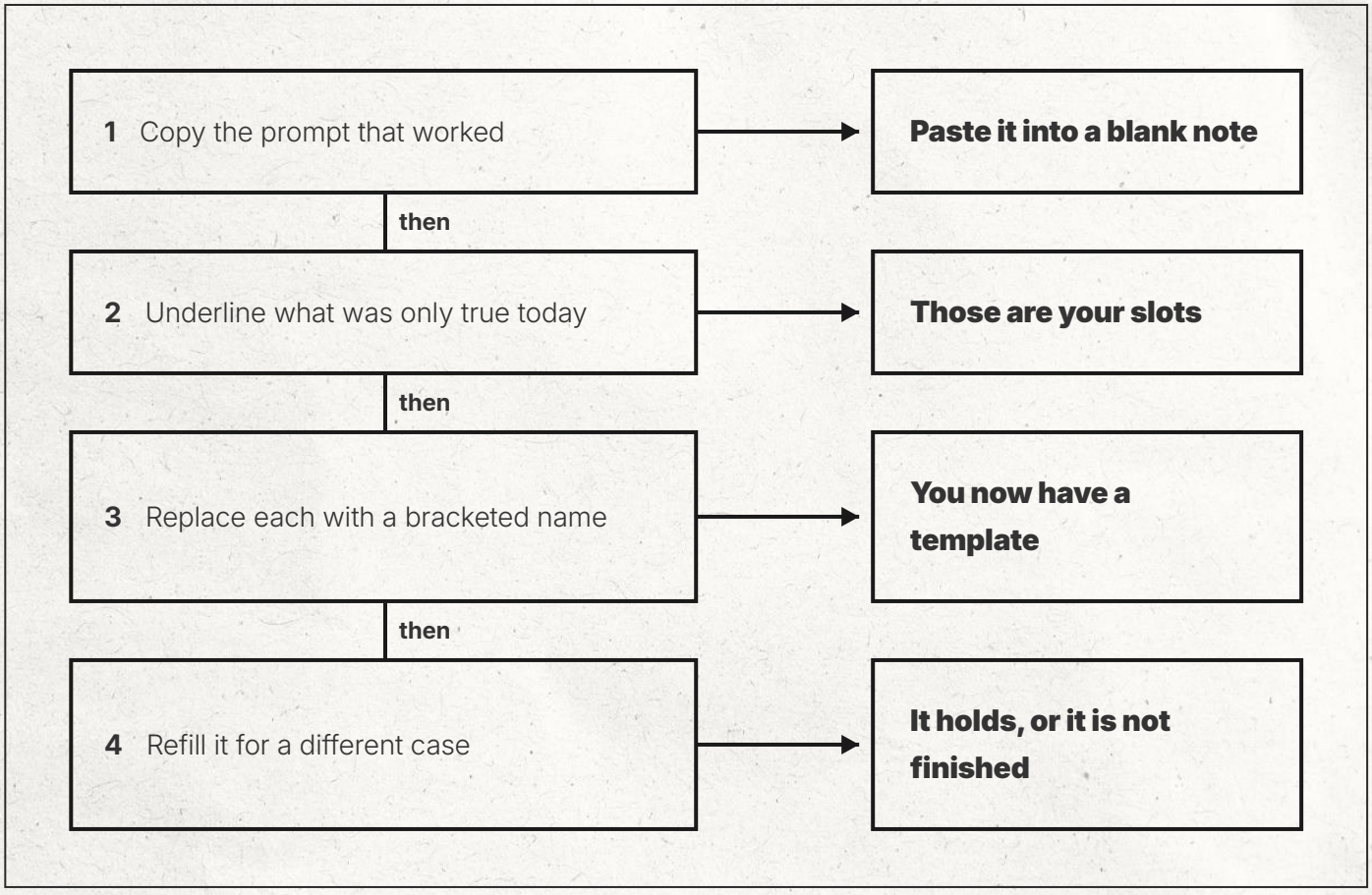


Fig 17.2 Converting one prompt into a template, in four passes. Step 4 is the one people skip, and it is the one that catches the detail you left hard-coded by accident.

17.2

Read it line by line. Ask of every sentence: would this be word for word identical next time?

Name the slot, not the value. [vendor name] tells the next person what goes there. [Harlow] tells them nothing.

The instructions are the asset. The part you were proudest of writing is almost always the fixed part.

Take a real one. Say you asked for this, and it gave you something you took into a meeting: "You are advising a chief operating officer. Below is a proposal from a logistics vendor called Harlow. We are deciding whether to move European returns handling to them in Q1. Pull out what they are actually committing to, what is conditional, and what is missing. Give me six bullets, then three questions I should ask before the next call. Keep it under 200 words and do not soften anything."

Now read it again and ask of each sentence whether it would be identical next time. The vendor changes. The document changes. The decision changes. Everything after that does not: the three things to pull out, the six bullets, the three questions, the word limit, the instruction not to soften. That second half is the part that took you three attempts to get right, and it is the part that transfers untouched to every proposal you will ever read.

So mark the four things that moved and leave the rest alone: "You are advising a [role]. Below is a [document type] from [source]. We are deciding whether to [decision]. Pull out what they are actually committing to, what is conditional, and what is missing. Give me six bullets, then three questions I should ask before the next call. Keep it under 200 words and do not soften anything." Square brackets are worth using because they are readable and no assistant treats them as special syntax. Say what goes in the slot rather than leaving an example sitting in it, because a leftover example is the single most common way a template gets sent out with somebody else's vendor name still in it.

That is the whole conversion, and notice how little of it is writing. You are not authoring a template. You are annotating a prompt you already wrote, which is why this takes about a minute and why it is worth doing at all.

Two habits make the result hold up. Add a short line at the top listing the slots to fill, so you cannot run it half-filled by accident. And if your prompt carried a worked example, the kind Chapter 14 taught you to include, keep the example inside the template. It is fixed, not swappable, and stripping it out to save space is how a template quietly stops producing what the original did.

Then run the test in step 4. Refill the template for a genuinely different case and send it. If the answer comes back wrong, something specific to the original job is still hiding in the fixed half, and you will find it in about ten seconds by comparing the two answers. A template you have used only once is a draft.

17.3 The moment to save a prompt is the moment it works

Not at the end of the day, and not when things calm down. The window is roughly 30 seconds wide, and it closes while you are still reading the answer.

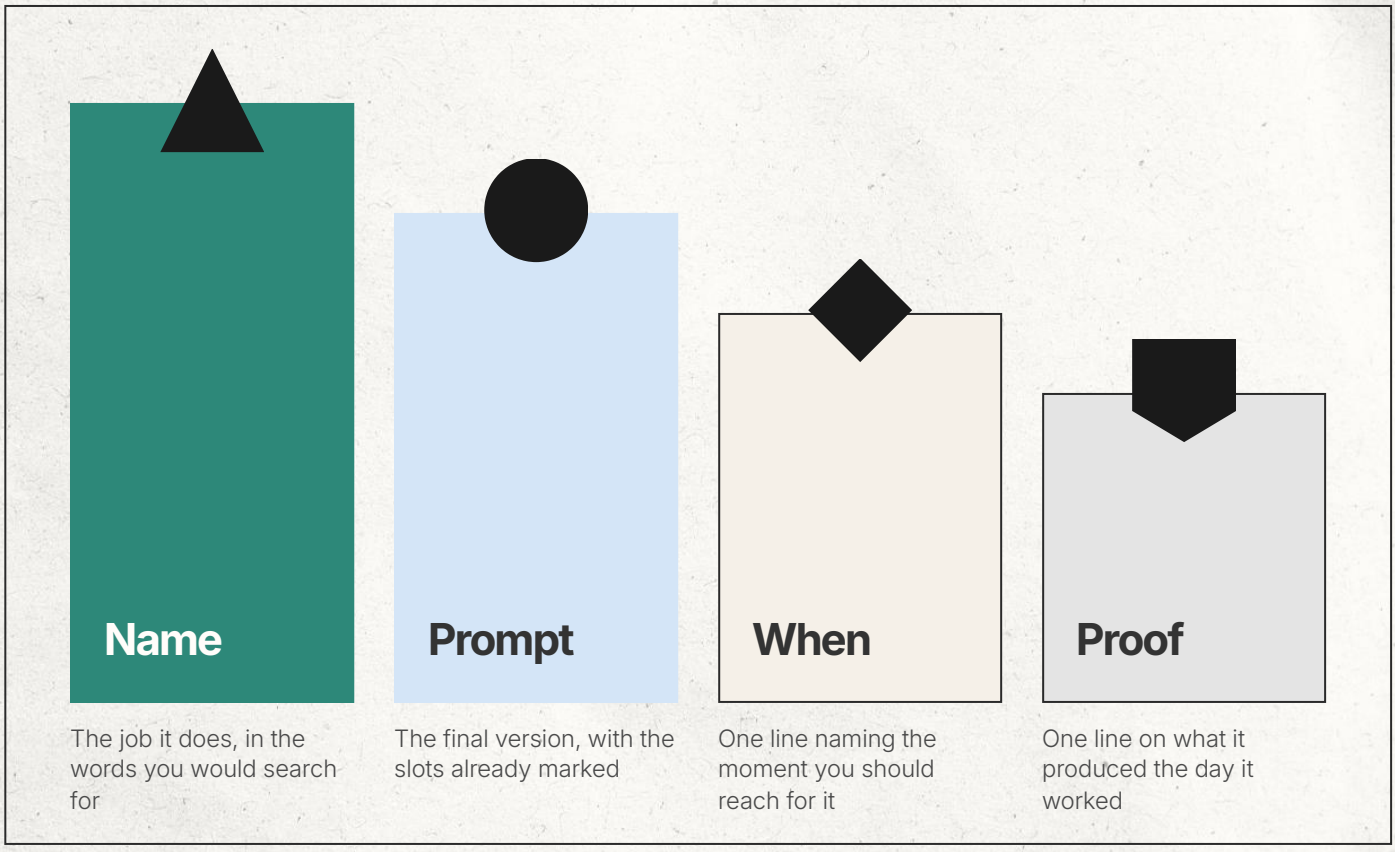


Fig 17.3 The four lines worth writing down. Bar heights are visual rhythm here, not a measurement: only the second line takes longer than a few seconds, and it is a copy and paste.

17.3

The trigger. The thought "that is exactly it" is the alarm. Save it before you use the answer.

Save the last version. The one that worked is the one after your corrections, not your opening message.

Earn the shelf. A prompt goes in the library the second time it works, not the first.

Everything in this chapter fails at one point, and it is not the writing. It is that nobody captures a prompt while it is still on screen. You read a good answer, you think that is exactly it, and then you use the answer, because using the answer is the thing you actually came to do. By the evening the prompt is under nine follow-ups and you could not pick the winning version out of the near-misses if you tried.

So make the trigger the reaction, not the calendar. The thought "that is exactly it" is the alarm going off. When you notice it, stop for half a minute and copy the prompt out before you touch the output. It feels like an interruption exactly once.

What you copy matters. Take the last version, not the first one you typed. The prompt that worked is your opening message plus every correction you made in the follow-ups, folded back in, and that folded-back version is the one carrying all the value. If the fix happened across three messages, write it as one prompt now while you still remember what the three fixes were.

Then add the three short lines around it. A name in the words you would actually search for. One line saying when to reach for it, which is the line that makes it findable by somebody who is not you and by you in four months. And one line on what it produced the day it worked, which is the only evidence you will ever have that this thing is worth keeping.

One piece of discipline stops the library rotting. A prompt earns its place the second time it works, not the first. Save it warm either way, but keep the second use as the bar for promoting it into the library proper, because a shelf full of prompts you used once is indistinguishable from a shelf full of prompts you will never use again.

Do this once today with the prompt you found in 17.1. The habit is built by the third repetition, and it never takes longer than the half-minute.

17.4 Keep them where you will actually look

All four assistants now offer somewhere to put a prompt. Most of what they offer is standing context rather than a template you invoke, and all of it loses to a plain document you already open every day.

Inside the assistant

A project, a Gem, a saved prompt: storage built into the tool you are already typing in.

TRAITS

· Zero copying: the prompt is where you use it

· Mostly stores standing context, not invocable templates

· What exists, and what it costs, differs sharply by assistant

PROS

+ Nothing to switch to at the moment of need

+ Files and knowledge can travel with the prompt

CONS

– Locked to one assistant, and awkward to share with colleagues

– Easy to forget the feature exists at all, which is fatal

A document you own

One file in the notes app or doc tool you already have open. One prompt per heading.

TRAITS

· Searchable with the search you already use daily

· Works identically no matter which assistant you switch to

· Costs one paste to use and one paste to add

PROS

+ Survives changing tools, jobs and pricing tiers

+ You can send a colleague the whole thing in one link

CONS

– Two extra clicks every time you use one

– Nobody maintains it for you, so it decays if you never prune

Fig 17.4 Where a template can live. The deciding question is not which is more capable, it is which one you will still be opening in month 3.

17.4

ChatGPT Custom instructions on every plan, Projects on free and paid, and custom prompts that its docs say appear in the composer as /prompts commands.

Claude Projects. Instructions written into a project apply to every conversation inside it, which is standing context rather than a template you call.

Gemini Gems: a name, instructions and optional knowledge files. Free with a personal Google account. Google says write the instructions as persona, task, context, format.

Copilot Saved prompts in Prompt Lab inside Copilot Chat: hover a prompt, Save prompt, and it lands under your saved prompts. Tied to your Microsoft 365 licence.

Look at those four notes together and you will spot something the marketing does not point out. Projects and Gems are containers you work inside, and instructions written into them apply automatically to everything you do there. That is Chapter 11's idea with a boundary drawn around it, and it is genuinely useful. It is not a template you reach for. Of the four, only Copilot's saved prompts and ChatGPT's custom prompts are really a shelf you take something down from.

Which is why the unglamorous answer wins more often than it should. One document, one prompt per heading, in whatever notes app or doc tool is already open on your second screen. It is searchable by the search you use forty times a day. It survives you changing assistant, changing employer, or losing a subscription tier. And you can send the whole thing to a colleague as one link, which no native feature on this list does gracefully.

Use the native feature where it is genuinely better, and it sometimes is. If you do the same job every week inside one assistant, a project or a Gem with the instructions written in saves you the paste, and Google's own advice for writing Gem instructions is persona, task, context and format, which is Chapter 13's framework under different names. Just be honest about the failure mode: a feature you have to remember exists is a feature you will stop using in three weeks, and the prompt you saved there is then more lost than if you had never saved it.

Naming is what makes the document work, and the rule is one line long. Name the prompt after the moment you need it, not after the technique it uses. "Vendor proposal to decision brief" gets found. "Summariser v3" does not, because in month 3 you will search for the word proposal and never for the word summariser.

Keep the whole thing to one file for as long as you possibly can. Folders feel organised and they add a decision every time you save something, and a decision at the moment of capture is precisely what a busy Thursday cannot afford. When one file genuinely stops working, split it once, by the kind of work rather than by the tool.

You are not building the finished system yet. Chapter 24 closes this track by assembling everything from AI 202 into a single playbook you can hand to somebody else. For now the job is smaller and more useful: get three templates written down and one place to keep them.

READING ENDS HERE

Everything before was reading. Everything after is doing.

3 exercises, About 15 minutes and one real decision that opens Chapter 18.

Do these before Chapter 18

3 exercises. About 15 minutes

Go back through your last two weeks of conversations and find one prompt that produced something you actually used. Write down the job it did, and the date, so you can find the thread again.

Split that prompt in two. On the left, every part that was only true for that specific job. On the right, the instructions that would be word for word identical next time.

ONLY TRUE THAT DAY	IDENTICAL EVERY TIME

Write the finished template, with each slot marked in square brackets. Then give it a name in the words you would search for, and write down the one place it now lives.

Chapter 17 on one page

TEMPLATE OR STANDING INSTRUCTION	Would you want it running on every conversation? Yes, it is a Chapter 11 standing instruction. No, and you still repeat it, it is a template you invoke
THE CONVERSION, IN 4 PASSES	Paste the prompt that worked · underline what was only true today · replace each with a [named slot] · refill it for a different case and check it holds
THE CAPTURE CARD	Name (searchable words) · the final prompt with slots marked · one line on when to reach for it · one line on what it produced
THE TRIGGER AND THE BAR	Capture the second you think "that is exactly it", before you use the output. Promote it into the library the second time it works
NAMING AND STORAGE	Name it after the moment of need, never the technique. One file, one prompt per heading, in a tool you already open daily. Split only when it truly stops working

BRING TO CHAPTER 18

Your three templates and the one place you keep them. Chapter 18 gives you the job most worth templating next, and you will want somewhere to put the prompt when it works.

BOTTOM LINE

You can now see the reusable shape inside a prompt you wrote once, mark its slots, capture it in the half-minute after it lands, and find it again months later.

NEXT CHAPTER

6 Questions to Ask Any Long Document

Chapter 18 turns the pile of documents nobody has time to read into something you can act on: summarising and extracting from long material without quietly losing the part that mattered.

BRING WITH YOU

- ☐ One template from this chapter, with its slots marked
- ☐ A long document you have been avoiding: a contract, a report, a deck

GO FURTHER

The Ultimate AI Library

Every guide, prompt pack and playbook AI Central has published, in one place. The reference shelf this course keeps pointing you to.

explore the library ↗



WHO MAKES THIS

AI CENTRAL

AI Central Media is a London-based editorial team. Our flagship publication covers practical AI for senior professionals, and 100+ AI companies, SaaS platforms and growth teams have advertised with us since 2023.

100+

AI companies, SaaS platforms, education brands and growth teams have advertised with us since 2023

613K

Accounts reached every month, across the 7 channels we operate

151

Countries our readers are in, plus all 50 US states

PROUD PARTNER OF

GAMMA · IIElevenLabs · guidde. · Outskill · Luma AI

HubSpot · attio · Synthflow · Notion · Replit

We turn attention into pipeline.

visit thecentral.ai ↗