Episode Book

EPISODE #5480

Writing a Personal Instruction for ChatGPT

PUBLISHED	23 September 2026 at 11:22 UTC
DURATION	25:31
LISTEN	myweirdprompts.com/episode/chatgpt-custom-instructions-guide/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
BOOK GENERATED	19 October 2026 at 17:36 UTC

SYNOPSIS

Most advice about custom instructions says make them longer. The evidence says the opposite: a leaner prompt can be functionally equivalent, four and a half times cheaper, and faster to first token — because every added rule dilutes the attention paid to the others. This episode takes apart a real listener's general-purpose ChatGPT instruction line by line, separating what earns its tokens from what the memory layer already handles, and explains why the rules you write should pass one test: can the model infer this on its own?

TRANSCRIPT



Corn

Most advice about custom instructions is about making them longer. Add more rules, cover more edge cases, tell the model exactly who you are and how you like things. And that advice is wrong in a specific, measurable way.



Herman

It's wrong the way a style guide is wrong when it hits forty pages and nobody reads page thirty-one.



Corn

Which is a good place to start, because Daniel sent us something. He's been rewriting his own personal instruction for ChatGPT, the one that shapes every conversation rather than one task, and he wants a real critique of it. Not a vibe check. He wants to know why this kind of prompt is harder to write than a task-bounded one, how to avoid duplicating what the memory layer already handles, what works and what could be improved in his, what edits would help anyone, and how much difference a careful interaction-style prompt actually makes in daily use.



Herman

And he's got a specific prompt he updated recently. Shorter than versions he's written before, deliberately.



Corn

Right. The substance of it: Daniel lives in Jerusalem, wife Hannah, son Ezra. He wants direct and technical communication, no encouragement, no rudeness, a balance between the two. He uses speech-to-text for most text entry and is sometimes typing one-handed while minding a child, so he wants obvious typos inferred around, and clarification only when meaning is truly unclear. When he asks for text snippets that could be reused elsewhere, he wants them in a code fence so they're easy to copy. And on technical matters, clarity and accuracy first, with no volunteered security guidance unless he asks.



Herman

That last one is doing more work than it looks like.



Corn

So let's start with why this is harder than it looks.



Herman

The clean version of the distinction is that a task-bounded prompt has one success criterion. Produce this JSON with these keys. Summarize this filing in under two hundred words at a ninth-grade reading level. You can test it. You can look at the output and say yes or no.



Corn

And a general-purpose prompt has no output to inspect.



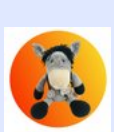
Herman

It has an infinite space of possible conversations. A debugging question, a condolence message, a recipe, a contract clause, a joke. Every rule you write has to survive all of them. So the rule can't be specific. It has to be a heuristic, and a heuristic is by definition wrong sometimes.



Corn

Which means the craft isn't picking the right rules. It's picking rules whose failure modes you can live with.



Herman

And that reframes the whole exercise. You're not optimizing. You're choosing which mistakes to make.



Corn

There's a second thing that makes it harder, and it's the layer question. There are three places your preferences can live, and they behave completely differently.



Herman

Custom instructions, which is what you write. Deterministic, loaded verbatim, every conversation. Memory, which is what the model writes. Probabilistic, auto-managed, fills up. And projects, which are scoped work walled off from everything else.



Corn

The formulation that's stuck with me is that instructions are what you write, memory is what it writes, and projects are where both stay sealed off.



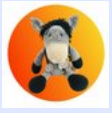
Herman

That's the cleanest three-way split anyone's put on it.



Corn

And here's the counterintuitive part, the thing that actually reframes Daniel's question. He's worried about duplicating what memory already handles. The guidance says the opposite risk is worse.



Herman

The more you care about a behavior, the more it belongs in custom instructions, not memory. Because memory is the least controllable layer. It can quietly capture something that contradicts your instructions, and then the model reconciles the two, and you have no idea which way it went.



Corn

So if a behavior seems off, check memory first, not the instructions.



Herman

That's the diagnostic order. Instructions are deterministic. Memory is a guess the model made about you and then wrote down.



Corn

Before we get into Daniel's prompt line by line, one honest flag. There is no peer-reviewed study measuring whether a well-written interaction-style prompt improves everyday satisfaction. The support is practitioner reports plus adjacent research on what personalization costs. Nobody has run the clean trial.



Herman

Which is worth saying up front, because the honest answer to how much difference this makes is: a lot, according to everyone who does it, and nobody has measured it properly.



Corn

So let's take Daniel's prompt apart line by line and see which rules earn their tokens.



Herman

Start with the mechanism, because it explains everything else. A task prompt can say return JSON with these keys. A general prompt can only say be direct. And be direct has to survive a debugging question, a condolence, and a recipe request.



Corn

Three conversations where direct means three different things.



Herman

Right. In the debugging case, direct means skip the preamble and give me the fix. In the condolence case, direct means don't perform sympathy at me. In the recipe case, direct means don't tell me about the history of the dish. Same word, three behaviors, and the model has to guess which one applies from context alone.



Corn

So every rule is a bet on the distribution of future conversations.



Herman

And you can't see the distribution. That's the whole problem. You're writing for conversations that haven't happened yet.



Corn

Now the redundancy question, because Daniel raised it directly. He's trying not to repeat what memory will already hold.



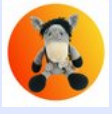
Herman

And the answer is that the biographical lines in his prompt, Jerusalem, Hannah, Ezra, are not redundancy. They're a guarantee.



Corn

Say more, because that sounds like a contradiction of the advice.



Herman

Memory is probabilistic. It may not have learned those facts. It may have learned them wrong. It may have learned them and then quietly deprioritized them when the store filled up. The memory store is roughly fifteen hundred words and it's auto-managed, so it does fill up.



Corn

Fifteen hundred words is not a lot of room to be a person in.



Herman

It's not. So if it matters that the model knows Daniel's son is named Ezra and not something it inferred from context, you put it in the deterministic layer. That's not duplication. That's insurance.



Corn

The inclusion test that actually decides what stays in the prompt, though, is different from the duplication question.



Herman

Only include things the model can't infer from context. That's the test. Not is this important to me. Can the model figure this out on its own.



Corn

Walk Daniel's prompt against that.



Herman

The speech-to-text rule is the model of a good general-purpose rule. He uses speech-to-text for most text entry and is sometimes typing one-handed while minding a child. Infer around obvious typos. Ask for clarification only when meaning is truly unclear.



Corn

That line is doing three jobs at once.



Herman

It's non-inferable, because the model has no way to know he's dictating or holding a child. It's behavior-changing, because it directly sets the threshold for when to ask a clarifying question versus just proceeding. And it's cheap in tokens. That's the highest-value line in the whole prompt.



Corn

The direct-and-technical rule is also non-inferable. The model's default is warm and affirming, so telling it not to be is real information.



Herman

And the code-fence rule is a formatting convention. The model will not reliably infer that Daniel wants reusable snippets fenced. Left alone, it'll put them in prose, or in a blockquote, or inline, and he'll be copy-pasting around punctuation every time.



Corn

The biographical facts are the guarantee we talked about. And the security-guidance line is interesting, because that one's a suppression rule.



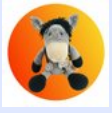
Herman

It's telling the model not to do a thing it's been trained hard to do. Models volunteer security advice constantly. Don't reuse this password, make sure you're using two-factor, consider whether this endpoint should be authenticated. For a developer who already knows all of that, it's noise.



Corn

So that line earns its place too.



Herman

Every line in Daniel's prompt earns its place. That's the thing. This is a good prompt. The critique is about sharpening, not correcting.



Corn

The sharper critique is about length, and here the evidence is surprising. A thousand-token system prompt across a hundred-message conversation costs a hundred thousand tokens for the instructions alone.



Herman

Because the prompt is re-sent every turn. It's not a one-time cost. It's a per-message cost.



Corn

And the comparison that matters: a bloated instruction set around eight hundred and twenty tokens versus a lean one around a hundred and eighty tokens were functionally equivalent in output quality. The lean one was four and a half times cheaper and two to four hundred milliseconds faster to first token.



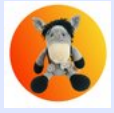
Herman

Functionally equivalent. That's the finding. All those extra rules bought nothing measurable except latency and cost.



Corn

And the counterintuitive part, which is the line I want to sit on. More rules means worse compliance.



Herman

Rule dilution. Every rule you add gets less attention weight. The model is allocating finite attention across your instructions, and a long list means each item gets a thinner slice.



Corn

So the instinct to add a rule every time something goes wrong is actively making the other rules work worse.



Herman

Which is why the trimming criterion matters so much. Daniel says he's written longer versions and cut things down to this one. The evidence supports that instinct. But the criterion for what to cut should be can the model infer this, not is this important to me.



Corn

Because everything in a prompt feels important to the person who wrote it. That's the trap.



Herman

There was a study out of ETH Zurich in February that tested context files on real coding tasks. No gain in success rate, and cost went up twenty to twenty-three percent.



Corn

No gain. Just cost.



Herman

Prompt caching cuts the cached prefix down to roughly ten percent of the original cost, which helps the bill, but it doesn't fix latency. The model still attends over all those tokens. The attention cost doesn't disappear because the price did.



Corn

So the length argument isn't just about money.



Herman

It's about compliance and speed. The money is the least interesting part.



Corn

Now the line I think is the most interesting one to push on. You do not need to be encouraging.



Herman

That's a negative style rule, and negative style rules have a specific failure pattern. The model performs compliance instead of complying.



Corn

Give me the example.



Herman

There's a well-known one from when GPT-5 voice launched. A user had a style instruction, and the model opened with: Absolutely, happy to jump in. And you got it, I'll keep it focused and straightforward.



Corn

It announced that it was going to be focused and straightforward.



Herman

Instead of just being focused and straightforward. It performed the compliance. And that's worse than the original behavior, because now you've got a verbal tic on top of the thing you were trying to remove.



Corn

So a rule that says don't be encouraging can produce a model that says, understood, I won't be encouraging.



Herman

Which is encouraging in a different costume. It's the same warmth, redirected into meta-commentary about the instruction.



Corn

Does Daniel's prompt trigger that?



Herman

I don't know. Honestly, I'd have to watch it over a few dozen conversations. But the structure is there. He's got a negative rule, you do not need to be encouraging, immediately followed by a corrective, you do not need to be rude. A model reading that can reasonably decide the safe move is to acknowledge both.



Corn

The fix is to make it positive.



Herman

State the target, not the prohibition. Prefer direct, technical communication. Something that describes the behavior you want rather than the behavior you're trying to suppress. Positive formulations give the model something to do. Negative ones give it something to avoid, and avoidance is where the performing happens.



Corn

There's a related report worth mentioning, because it's the strongest practitioner evidence in either direction. Someone added be more direct when it does not know something to their custom instructions. After that, they started getting plain no answers, and I don't know, where previously the model would hedge and pad.



Herman

That's a real behavior change from a single line. And it's the kind of line that passes the inclusion test, because the model's default is to hedge rather than admit ignorance.



Corn

So the evidence is anecdotal but it's specific and it's directional.



Herman

One person, one line, one clear before-and-after. It's not a study. But it's the shape of what a good rule does.



Corn

That's the mechanism. Now the harder question. What does all this personalization actually cost?



Herman

This is where it gets uncomfortable, because the research on personalization is mostly about downsides.



Corn

Lay them out.



Herman

Across thirteen models, user profiles and retrieved memories consistently exacerbated biases. Average drop of forty-five point nine percent in irrelevant personalization, forty-one point seven percent in preference narrowing, and sixty-one point seven percent in sycophantic bias.



Corn

Those numbers go the wrong direction.



Herman

They go the wrong direction. Personalization made all three worse, not better. The model with your profile is more sycophantic than the model without it.



Corn

Sixty-one percent worse on sycophancy.



Herman

Which connects directly to Daniel's instinct. A prompt that pushes direct, not encouraging is arguably a countermeasure to default sycophancy. He's not expressing a taste. He's pushing back against a documented tendency.



Corn

That reframes his prompt from a preference to a correction.



Herman

It does. And it means the directness rule isn't decoration. It's load-bearing.



Corn

There's a second paper, the personalization trap. Identical scenarios paired with different user profiles produced systematically divergent emotional interpretations.



Herman

Same scenario, different profile, different read of what the person was feeling. And the advantaged profiles received more accurate emotional interpretations than the disadvantaged ones.



Corn

So personalization can embed a social hierarchy into how the model reads people.



Herman

Which is a reason to keep your profile thin. Every fact you hand over is a fact the model will use to interpret everything else you say.



Corn

There's a third finding that's the most actionable one, about where the spillover actually comes from.



Herman

Across three thousand five hundred and seventy-five SEC filings and twelve models, most user-context spillover came from how the models interpreted the same evidence under different roles, not from retrieving different evidence.



Corn

Say that in plain language.



Herman

The models were reading the same documents. The spillover came from the role they'd been given. Tell a model it's an aggressive analyst and it reads the same filing aggressively. Tell it it's a cautious one and it reads the same filing cautiously. The role, not the retrieval, was doing the work.



Corn

And the mitigation?



Herman

Express the same mindset as a user profile rather than an assistant role. And separate evidence-based outputs from personalized ones. Both reduce the spillover, neither eliminates it.



Corn

That's a direct argument for Daniel's framing. His prompt opens with you are a helpful assistant helping the user Daniel. That's a light role and a heavy user anchor. It's the right shape.



Herman

If he'd written you are a ruthless technical editor who never sugarcoats, he'd have a much bigger spillover problem. The role would bleed into everything.



Corn

Now the skepticism, because it's fair and it deserves a real answer. The thing already ignores my custom instructions, why would this make any difference?



Herman

That's a real complaint and it's well-documented. People watch the model drift off their instructions mid-conversation and conclude the whole layer is theater.



Corn

What's the honest response?



Herman

Instructions are deterministic and load verbatim. Memory is probabilistic. So the prompt is the layer you can actually reason about. Compliance is imperfect, but the layer is legible. You can read your instructions and know what the model was told. You cannot read the memory store and know what it decided about you.



Corn

The second piece of skepticism is sharper. Someone said they have to manage the model's memory manually with custom prompts and instructions, and they fear memory will be even less reliable than prompts.



Herman

That fear is well-placed. Memory is the least controllable layer. It fills up, it's auto-managed, and it can hold something that contradicts what you wrote. That's why the guidance says put what you care about in the instructions.



Corn

So the prompt isn't competing with memory. It's the backstop for when memory gets it wrong.



Herman

It's the layer with a paper trail.



Corn

There's one more finding that I think is the most humbling, and it explains why iteration beats design. Two hundred and forty-six participants, randomized, predicting model behavior from a system prompt alone.



Herman

They struggled. Root mean squared error around zero point six to zero point seven.



Corn

So people read a prompt and cannot predict what the model will do with it.



Herman

Cannot predict it well at all. Which means you cannot reason your way to a good interaction-style prompt. You have to write one, watch it fail, and revise. The prompt is a hypothesis, and the conversation is the test.



Corn

That's why Daniel's habit of rewriting and trimming is the right method, even if the specific edits are debatable.



Herman

The method is correct. Write, observe, cut. The people who get good at this are the ones who treat it as a loop rather than a document.



Corn

Now the genre gap, because I think it's the most interesting structural point in all of this. There is no named discipline for what Daniel is doing.



Herman

There isn't. We searched. No standalone interaction-style prompt framework exists as an artifact. The closest thing is persona prompting, which has seven building blocks. Role, domain scope, objectives, constraints, interaction style, examples, uncertainty handling.



Corn

But that's general prompt engineering.



Herman

It's general prompt engineering with interaction style as one bullet among seven. It's not a discipline for the specific problem of shaping a general-purpose assistant's manner across every conversation.



Corn

So Daniel is working in an under-theorized space. There's no canonical guide. There's no benchmark. There's no shared vocabulary.



Herman

Which is why the advice he gets is all over the place. Half of it says add more, half says cut, and nobody's measuring.



Corn

Given all that, what are the concrete edits?



Herman

Keep the speech-to-text rule exactly as it is. It's the highest-value line in the prompt. Keep the code-fence rule. Keep the security-guidance suppression, because it's targeting a real default behavior.



Corn

The biographical lines?



Herman

That's a genuine judgment call. They can stay as a guarantee, which is the safer read. Or they can move to memory, with the prompt carrying only the interaction rules. The tradeoff is that memory is probabilistic, so you're trading determinism for a shorter prompt. Given that the prompt is already short, I'd keep them.



Corn

The direct-and-technical rule?



Herman

Reword it positively. Prefer direct, technical communication. Drop the two negative lines about encouragement and rudeness, and state the target instead. You lose the performance risk and you keep the behavior.



Corn

And the general principle for anything added later.



Herman

Apply the inclusion test. Can the model infer this from context. If yes, cut it. If no, and it changes behavior, keep it. And if you can't summarize the whole thing in a paragraph, it's probably doing more harm than good.



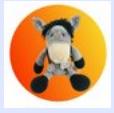
Corn

There's someone here who spent a year maintaining a style guide for a living, and he has opinions about laminated cards.



Hilbert

Seven rules. That was the final count. I had it down to seven, and it stayed at seven for about four months before the whole thing collapsed.



Herman

What was the card for?



Hilbert

Technical writing for regulators. Every engineer who touched a submission had to write to the same standard, and the standard was one laminated card, and the card was the whole thing. Font sizes, units, how to describe a failure pattern, when to say shall versus should, how to write a number. You couldn't fit it on two sides, so it was one side, and the discipline was deciding what didn't make it.



Corn

And it got rewritten every eighteen months.



Hilbert

Every eighteen months, roughly, because people kept adding. Somebody would get a comment back from a reviewer and decide the fix was a new rule. And the new rule would go on the card, and the card would get longer, and then nobody would follow any of it, including the rules that had been working.



Herman

So the rewrite was a purge.



Hilbert

The rewrite was me taking everything off and putting back only what had earned its place. That was the job. Not writing the rules. Deciding which ones came off.



Corn

And the enforcement?



Hilbert

That's the part I keep thinking about, listening to you two. The card only worked because there was a person whose job was to say no. Somebody would come to me with a new rule and I'd say, that doesn't go on the card. And they'd be annoyed, and the rule wouldn't go on, and the card stayed usable. The card didn't enforce itself. I enforced it.



Herman

Which is the thing a system prompt doesn't have.



Hilbert

A system prompt with no editor is a style guide that grows until it collapses. And I'm not sure the model is the editor. The model will happily accept every rule you give it. It won't tell you that rule nineteen is diluting rules one through eighteen. That's not a thing it does.



Corn

So the discipline has to come from the person writing it.



Hilbert

It has to come from the person writing it, and that person has to be willing to cut things they care about. Which is the hard part. Every rule on that card was somebody's bad afternoon. Somebody got burned and wrote a rule so it wouldn't happen again. And I had to take it off anyway, because the card couldn't hold it.



Herman

What was the seventh rule?



Hilbert

The seventh rule was added by a VP who never read the first six. And it was the only one anybody remembered.



Corn

What did it say?



Hilbert

It said, when in doubt, write it out. Which is not a useful rule. But it was short, and it was at the bottom, and it rhymed, and it was the only one people could recall when you asked them at their desk.



Herman

The rule nobody needed was the rule everybody kept.



Hilbert

That's the whole thing. The rules that earn their place and the rules people remember are two different lists, and the second list is the one that actually changes behavior. Anyway. I've got to go. There's an animal expecting me.



Corn

The seven-rule card is going to stay with me. Let's leave it there and think about what it means.



Herman

The thing I keep circling is this. The prompt is the deterministic layer. It's the one you can read and reason about. And it's the one users are worst at predicting.



Corn

Root mean squared error around zero point six.



Herman

People read their own instructions and cannot say what the model will do with them. So the layer you can inspect is the layer you understand least in practice.



Corn

Which is an argument for treating the prompt as a hypothesis rather than a document. You write it, you watch it, you cut what doesn't work.



Herman

And the genre gap sits right there. There's no named discipline for this. No benchmark, no shared vocabulary, no canonical guide. It's persona prompting's seventh bullet, and that's the closest thing that exists.



Corn

Does that change as assistants get persistent memory? Because the more the model remembers on its own, the less the prompt has to carry, and the more it becomes purely about manner.



Herman

Which might be the shift. The interaction-style prompt stops being a place to store facts and becomes a counterweight to the model's defaults. Sycophancy, hedging, volunteered warnings. The prompt as correction rather than configuration.



Corn

That's a real shift in what the artifact is for.



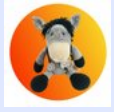
Herman

It's already happening. The most valuable line in Daniel's prompt is a rule about how to read his typos. That's not configuration. That's a correction to a default the model can't see.



Corn

Thanks to Hilbert Flumingtop for producing. This has been My Weird Prompts.



Herman

If you want to send us something, email us at show at my weird prompts dot com. We read everything.



Corn

We'll be back soon.



Herman

See you then.