

Belt and Braces.

How to actually use the AI Tells cheat sheet and the ai-tells repo-based skill.

The system prompt is the belt.

The skill file is the braces.

Both together = your trousers stay up.

By the end of this guide you will:

1. Have an AI Signs system prompt installed (the belt)
2. Know the repo-first way to use the skill (the braces)
3. Know how to run the workflow on any draft
4. Know which tool to use for which kind of writing
5. Have a maintenance schedule so the list doesn't date

Companion files in today's pack:

- AI Tells Cheat Sheet (PDF) - the compressed list
- ai-tells GitHub repo - canonical SKILL.md + install notes

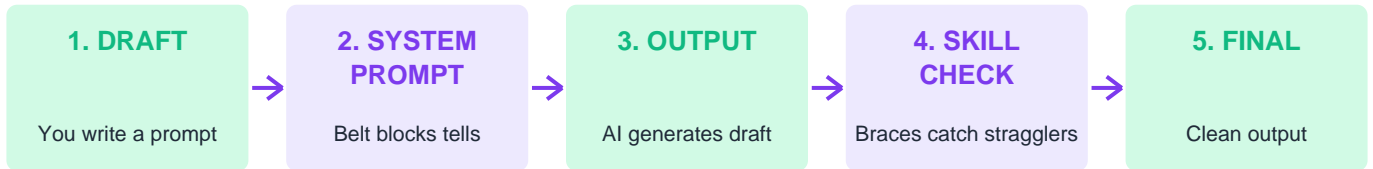
<https://github.com/kdgbalmer/ai-tells>

By Kyle Balmer

AI with Kyle | aiwithkyle.com

The Workflow

Two layers. The first one stops most AI tells from being generated. The second catches anything that slipped through. Both at once is the strongest setup. Either alone still helps.



Layer 1 vs Layer 2

BELT (Layer 1)

Before you write

- System prompt / project instructions
- Cheat sheet pasted in once
- Active on every chat in the project
- Stops most tells being generated

BRACES (Layer 2)

After you write

- Repo-based skill invoked on a draft
- Audits the output
- Flags anything that leaked
- Suggests human alternatives

Installing The Belt

Layer 1: a system prompt that blocks AI tells before they're generated.

If you use Claude Projects

1. Open claude.ai and go to your Project (or create a new one)
2. Click 'Edit instructions' on the right sidebar
3. Paste the system prompt template (next page) into the box
4. Save. From now on, every chat in this project follows it.

If you use ChatGPT

1. Open chatgpt.com, click your profile, then 'Customize ChatGPT'
2. Paste the system prompt into the second box ('How to respond')
3. Save. Active on every new chat from now on.

If you use Claude Code / Codex / Cursor

1. Create or open CLAUDE.md / AGENTS.md in your project root
2. Add the system prompt template as a new section
3. Commit it. The agent reads it on every session.

Tip: don't just paste the cheat sheet word-for-word. Trim it to the categories that matter most for your writing. Marketing copy needs the promotional tier. Documentation needs the structural tier. Tailor it.

Belt: System Prompt Template

Paste this into Claude Project instructions, ChatGPT custom instructions, or CLAUDE.md.

AVOID THE FOLLOWING WHEN WRITING:

Vocabulary: delve, tapestry, multifaceted, pivotal, intricate, robust, vibrant, meticulous, nuanced, leverage, foster, navigate, underscore, showcase, ensure, realm, garner, bolster, enduring, elevate, testament, journey, landscape, ecosystem, paradigm, pivot, harness.

Structures: 'It's not X, it's Y' (negative parallelism), bullet points for short answers, markdown headers on quick replies, rule-of-three triads, outline-like conclusions (Despite challenges...), Title Case section headings.

Openers: 'Great question', 'Absolutely', 'Certainly', 'I'd be happy to help', 'It's important to note', 'Furthermore', 'Moreover', 'In conclusion'.

Punctuation: em dashes at every clause boundary, curly quotes in plain text, ellipses on every other line.

WRITE INSTEAD WITH:

- Short declaratives at paragraph beats
- Plain section headings (sentence case)
- Specific numbers, not round ones
- Concrete examples, not abstract claims
- Opinion landed flat, no hedging
- Voice consistent with: [PASTE YOUR VOICE GUIDE HERE]

Customise the [VOICE GUIDE] line

Replace the bracketed bit with a description of your own voice. Examples: 'British, casual, opinionated' or 'plainspoken American with dry humour' or '5-sentence paragraphs maximum, conversational'. The blocklist alone produces neutral output - the voice line gives it shape.

Installing The Braces

Layer 2: a repo-based skill you invoke on any draft to catch what leaked through.

Method 1: In a normal chat

1. Open the ai-tells repo and copy the current SKILL.md / prompt block
2. Paste it into Claude, ChatGPT, or Codex as the instruction layer
3. Add your draft underneath
4. Ask for: audit, flagged tells, and a clean rewrite
5. Best for one-off checks or testing the workflow quickly

Method 2: In Claude Projects

1. Open your writing project in Claude
2. Put the cheat-sheet belt in Project Instructions
3. Add the repo-based skill instructions as project knowledge or a skill
4. Reuse that project for newsletters, posts, and sales copy

Method 3: In Claude Code / local skills

Clone or open the ai-tells repo, then place the skill in your normal local skills flow. In Claude Code, Codex, or Cursor, keep the belt in CLAUDE.md / AGENTS.md and call the skill when a draft needs a final AI-tells pass.

Canonical repo:

github.com/kdgbalmer/ai-tells

Use that as the source of truth instead of an old local generated file.

Example Walkthrough

Here's the full workflow on a real prompt. Watch the input and output evolve as each layer is added.

STEP 1: NO BELT, NO BRACES

Prompt: 'Write a 3-line LinkedIn post about local LLMs.'

In today's rapidly evolving AI landscape, local LLMs are emerging as a transformative solution that empowers creators to leverage cutting-edge models without relying on the cloud. It's not just about saving money - it's about reclaiming control. Embrace the shift.

STEP 2: WITH BELT (system prompt blocking tells)

Local LLMs run on your laptop instead of someone else's servers. Your prompts never leave your machine. The model is free after the download. It's not a Claude replacement, but for routine work it's surprisingly capable.

STEP 3: WITH BELT + BRACES (skill caught one tell)

Skill flagged: 'It's not a Claude replacement' (negative parallelism). Suggested rewrite below.

Local LLMs run on your laptop instead of someone else's servers. Your prompts never leave your machine. The model is free after the download. Won't replace Claude for hard tasks. Will do most routine work just fine.

When To Use Which Layer

Not every piece of writing needs both layers. Pick based on stakes.

Internal Slack message

Neither. Don't bother. Effort doesn't match stakes.

Email to a colleague

Belt only. The system prompt is doing all the work in the background.

Draft of a LinkedIn post

Belt + light scan. Run your eye over the cheat sheet for 30 seconds.

Blog post / newsletter

Belt + Braces. Mandatory. This is the use case the workflow was built for.

Sales page / landing copy

Belt + Braces + manual edit. Stakes are high, AI tells kill conversion.

Book chapter / long form

Belt + Braces + full human rewrite. AI is a starting point only.

Anything legal or contractual

AI for first draft, professional human review mandatory.

The point isn't to make all AI writing undetectable. It's to match the polish to the audience. A reply to your accountant doesn't need Hemingway. A sales page absolutely does.

Maintenance & What Next

AI tells shift as models are updated. The list dates. Schedule a 30-minute review every quarter.

Quarterly review checklist

- Open Wikipedia: Signs of AI writing - check what's been added
- Open the OpenAI / Anthropic release notes - check for behaviour changes
- Read 5 of your own AI-assisted drafts - spot anything new and consistent
- Update the cheat sheet word lists - add new tells, demote trained-out ones
- Update the system prompt template in your projects
- Pull the latest ai-tells repo version and check SKILL.md changes
- Re-download or re-generate your local copies

What this workflow does NOT replace

- It does not replace having something to say. AI tells removed from a hollow draft = a slightly less obvious hollow draft.
- It does not replace voice. The blocklist produces neutral. Voice is additive.
- It does not replace human reading at the end. Catches surface tells, not bad reasoning.

Daily AI breakdowns at aiwithkyle.com

Newsletter: free, daily, plain English.

YouTube: AI with Kyle - full livestream replays.

Skill repo: github.com/kdgbalmer/ai-tells

Belt and braces. Stop sounding like a robot. - Kyle