M A V E R I C K  A I

# AI CANARIES

## HOW TO CATCH WHEN YOUR AI STARTS LYING TO YOU

*AI does not go from perfect to broken overnight. It drifts. This guide shows you how to detect the drift before you start trusting outputs that are completely wrong.*

What canaries are and where the concept comes from

Why AI sessions degrade over time

The exact degradation timeline (what breaks and in what order)

How to set up canaries in Claude, ChatGPT, and other tools

7 types of canaries you can use right now

What to do the moment your canary dies

Advanced strategies for long sessions and production use

2026 Edition

# What Is a Canary?

In the 1800s, coal miners had a problem. Toxic gases like carbon monoxide and methane would build up in mine shafts, and by the time anyone noticed, it was often too late. The gases were invisible and odorless. So miners started bringing canaries underground with them.

Canaries are extremely sensitive to toxic gases. If the air went bad, the bird would stop singing and collapse before the gas reached levels that were dangerous to humans. When the canary dropped, miners knew to get out immediately. The bird was not the point. The bird was the signal.

That same idea has been borrowed all over technology. In software, a canary deployment is when you roll out a change to a small percentage of users first. If something breaks for that small group, you catch it before it hits everyone. The small group is your canary.

## Canaries in AI

When you work with an AI model like Claude or ChatGPT, the same principle applies. You add a small, easy-to-check instruction to your prompt. Something the model should always follow if it is paying attention. Then you watch whether it keeps following it.

Peter Steinberger, the creator of OpenClaw (one of the largest open-source AI agent platforms), popularized this technique for Claude. The idea is simple: add a tiny rule like "always start your response with my name" to your system prompt or CLAUDE.md file. As long as Claude keeps doing it, you know the session is healthy. The moment it stops, something has shifted.

| 1 | 100% | 0 sec |
|---|---|---|
| Small instruction | Signal accuracy | Detection time |

**THE CORE IDEA**

A canary is a trivially simple instruction that acts as a health check for your AI session. If the model cannot follow a one-line rule, it definitely cannot handle your complex instructions either.

# Why AI Sessions Degrade

Before you can use canaries effectively, you need to understand why AI sessions go bad in the first place. It is not random. There are specific, predictable reasons why a model that was perfect at the start of a conversation becomes unreliable later.

## Context window limits

Every AI model has a maximum context window. Think of it like working memory. Claude Sonnet has roughly 200,000 tokens of context. That sounds like a lot, and it is. But here is the thing: not all of that context gets equal attention.

Research on large language models has consistently shown that information at the very beginning and very end of the context window gets the most attention. Information in the middle tends to get less weight. This is sometimes called the "lost in the middle" problem. As your conversation grows longer, your original instructions (including your canary) end up further and further from the edges, sitting in that low-attention zone.

## Instruction dilution

When you start a conversation, the model has maybe one or two messages to focus on. Your instructions are front and center. Fifty messages later, those original instructions are buried under thousands of tokens of conversation. The model is now juggling your initial rules, your ongoing conversation, all of its previous responses, and whatever new context you just added. Something has to give.

The instructions that give first are usually the smallest, most peripheral ones. A rule like "always use my name" does not feel important compared to "answer my question about database architecture." So the model unconsciously deprioritizes it. That is exactly why it makes such a good canary.

## Attention drift

AI models process your entire conversation every time they generate a response. As the conversation gets longer, the model spreads its attention across more and more content. Your system prompt was 100% of the context in message one. By message fifty, it might be 2% of the context. The model is not ignoring your instructions on purpose. It is just spreading itself thin.

## Compounding errors

If the model makes a small mistake in message 10 and you do not catch it, that mistake becomes part of the conversation history. Now the model treats its own wrong answer as established fact and builds on top of it. Each subsequent response can drift a little further from reality. Without a canary, you might not notice until the outputs are completely off.

**HEADS UP**

AI degradation is not sudden. You will not get a warning or an error message. The quality just quietly slides. That is what makes it dangerous, and that is what makes canaries so useful.

# The Degradation Timeline

AI sessions do not fail randomly. They follow a pattern. Understanding this pattern is the whole reason canaries work. Here is what typically happens, in order.

## STEP 1: The Honeymoon Phase (Messages 1-15)

Everything works. The model follows every instruction perfectly. Your canary fires on every response. Formatting is clean. Tone is right. Answers are accurate. This is where most people form their baseline expectations of the model.

## STEP 2: The Subtle Slip (Messages 15-40)

The model starts making small, easy-to-miss changes. Maybe it shortens its responses slightly. Maybe it stops using a specific formatting rule you set. Maybe it starts paraphrasing your terminology instead of using your exact words. Nothing is wrong per se. Things are just a little less precise.

> **TIP**
>
> This is the stage where most people do not notice anything. They assume the model is still running at full quality. A canary catches it here.

## STEP 3: The Canary Dies (Messages 30-60)

Your canary instruction gets dropped. The model stops using your name, or stops formatting responses the way you asked, or skips the sign-off you required. This is your signal. The model is no longer reliably tracking your instructions.

> **HEADS UP**
>
> The exact message count varies depending on the model, the length of your messages, and how much context you are feeding in. A conversation with short messages might last 80 turns before the canary dies. A conversation with long code blocks might lose the canary after 15 messages.

## STEP 4: The Drift Zone (Messages 40-80)

If you keep going past a dead canary, the model starts making assumptions it should not make. It fills in gaps with plausible-sounding information that might be wrong. It stops asking clarifying questions. It starts treating its own earlier mistakes as facts. The outputs still look confident and well-written, but the underlying accuracy is sliding.

## STEP 5: Full Hallucination (Messages 60+)

At this point the model is confidently generating information that is partially or completely fabricated. It might invent function names that do not exist, cite papers that were never published, or give you technical

explanations that sound authoritative but are wrong. The problem is that the tone does not change. The model sounds just as confident as it did in message one.

| Stage | What Happens | Canary Status | Risk Level |
|---|---|---|---|
| Honeymoon | Everything works perfectly | Alive and singing | None |
| Subtle slip | Small formatting/precision changes | Still alive | Low |
| Canary dies | Small instructions get ignored | Dead | Medium |
| Drift zone | Assumptions, shortcuts, gaps | Long dead | High |
| Hallucination | Confident but wrong outputs | Forgotten entirely | Critical |

# The Name Canary

The most popular and simplest canary is the name canary. You tell the model to start every response with your name. That is it. One line. No complexity. And it works remarkably well.

## Why the name trick works

Using your name at the start of every response is a perfect canary because it checks three things at once. First, can the model still access your system prompt or initial instructions? Second, can it follow a simple formatting rule? Third, is it still prioritizing your custom instructions over its default behavior?

The model does not naturally want to start every response with your name. It has no built-in reason to do it. So when it does, that means your custom instructions still have weight. When it stops, those instructions are losing influence.

## How to set it up

CLAUDE.MD / SYSTEM PROMPT

```
# Canary

Always begin every response with my name: [Your Name].
This is a session health check. If you ever stop
doing this, I know the session is degrading and
it is time to start fresh.
```

CHATGPT CUSTOM INSTRUCTIONS

```
# How would you like ChatGPT to respond?

Start every response with my name: [Your Name].
Do not skip this under any circumstances.
```

## What it looks like in practice

Here is what a healthy session looks like versus a degraded one.

HEALTHY SESSION (CANARY ALIVE)

```
You: How do I set up a webhook in Stripe?

Claude: Hey Sarah, to set up a webhook in
Stripe, head to the Developers section in
your dashboard and click Webhooks...
```

DEGRADED SESSION (CANARY DEAD)

```
You: How do I set up a webhook in Stripe?

Claude: To set up a webhook in Stripe, go
to the Developers section and click on
Webhooks. You will want to...
```

**TIP**

Notice the answer in the degraded session might still be correct. That is the tricky part. The information could be fine for another few messages. But the canary is telling you the model is no longer reliably tracking your instructions, and the accuracy will start slipping soon.

# 7 Types of Canaries You Can Use

The name canary is the most popular one, but it is not the only option. Different canaries test different things, and combining multiple canaries gives you a more complete picture of session health.

## 1. The name canary

**Instruction:** "Always start your response with my name."
**What it tests:** Whether your system prompt is still getting attention.
**Best for:** General use, Claude.md files, ChatGPT custom instructions.

## 2. The sign-off canary

**Instruction:** "End every response with a horizontal rule and the word 'Ready' on a new line."
**What it tests:** Whether the model can follow formatting rules through to the end of its response. This is harder than the name canary because the model has to remember the rule through its entire generation, not just at the start.
**Best for:** Coding sessions, long-form writing, complex multi-step tasks.

## 3. The formatting canary

**Instruction:** "Always use bullet points, never numbered lists" or "Always bold the first sentence of every paragraph."
**What it tests:** Whether the model is still following your style rules. Formatting is one of the first things to drift because the model has its own default formatting habits that start taking over as your instructions lose weight.
**Best for:** Content creation, report writing, any work where output format matters.

## 4. The language canary

**Instruction:** "Never use the word 'certainly' or 'absolutely' in your responses" or "Do not use em dashes."
**What it tests:** Whether the model can suppress its default habits. AI models have favorite words and patterns. Telling it to avoid specific ones tests whether your instructions still override the model's defaults.
**Best for:** Writing tasks, brand voice work, editorial style enforcement.

## 5. The role canary

**Instruction:** "You are a senior backend engineer. Always reason through problems from a systems architecture perspective first."
**What it tests:** Whether the model is still in character. When a role canary dies, the model reverts to its generic helpful assistant persona. You will notice it starts giving more surface-level answers instead of the deep, role-specific reasoning you set up.
**Best for:** Expert consultation, specialized research, roleplay-driven work.

## 6. The constraint canary

**Instruction:** "Keep all responses under 200 words" or "Always include exactly 3 action items."
**What it tests:** Whether the model can follow hard constraints. This is a strong canary because it is easy to verify. Either the response is under 200 words or it is not. Either there are 3 action items or there are not.
**Best for:** Structured output, automated pipelines, situations where precision matters.

## 7. The meta canary

**Instruction:** "At the end of every response, rate your confidence from 1-10 that you followed all of my instructions in this conversation."
**What it tests:** Whether the model is still self-aware about your instructions. This is the most advanced canary because it forces the model to actively check itself. When the meta canary dies, the model has fully lost track of your original rules.
**Best for:** Research tasks, high-stakes work, situations where accuracy is critical.

| Canary Type | Difficulty to Follow | Sensitivity | Best Use Case |
|---|---|---|---|
| Name | Very easy | Medium | General use |
| Sign-off | Easy | Medium-high | Coding, writing |
| Formatting | Medium | High | Content, reports |
| Language | Medium | High | Writing, brand voice |
| Role | Hard | Very high | Expert consultation |
| Constraint | Medium | Very high | Structured output |
| Meta | Very hard | Highest | High-stakes research |

# Setting Up Canaries in Different Tools

The concept is the same everywhere, but the setup differs depending on which AI tool you are using. Here is how to add canaries to each one.

## Claude (claude.ai, Desktop App)

### STEP 1: Open Project Settings

Go to claude.ai and create a new Project (or open an existing one). Click the project settings gear icon. You will see a field called "Custom Instructions." This is where your canary goes.

### STEP 2: Add Your Canary Instruction

**PROJECT CUSTOM INSTRUCTIONS**

```
Always start your response with 'Hey [Name],'

This is my canary instruction. If you ever
stop doing this, I need to know the session
is degrading.
```

### STEP 3: Verify It Works

Send a test message. Confirm the model uses your name. Then continue your normal work and keep an eye on it.

## Claude Code (Terminal)

Claude Code uses a file called CLAUDE.md in your project root. This file acts as a persistent system prompt that Claude reads at the start of every session.

**CLAUDE.md FILE**

```
# Session Rules

## Canary
Always start every response with my name:
[Your Name]. This is a health check for the
session. Do not skip it.

## Other Rules
- Use TypeScript, not JavaScript
- Run tests after every file change
- Keep functions under 50 lines
```

> **TIP**
>
> In Claude Code, the canary is especially useful because sessions tend to be long. You might be working on a codebase for hours. The canary tells you when it is time to start a new session with /clear or by opening a new terminal.

## ChatGPT

### STEP 1: Open Custom Instructions

Click your profile icon in ChatGPT, then go to Settings. Find "Personalization" and click "Custom Instructions." There are two text fields.

### STEP 2: Add the Canary

In the "How would you like ChatGPT to respond?" field, add your canary instruction. This will apply to every new conversation.

**CHATGPT CUSTOM INSTRUCTIONS**

```
Always start every response with my name:
[Your Name].

Never use the words 'certainly', 'absolutely',
or 'I would be happy to'.
```

## Other tools (Gemini, Copilot, local models)

Any tool that lets you set a system prompt supports canaries. The principle is the same: add a small, verifiable instruction before your main prompt. For local models running through Ollama, LM Studio, or similar tools, add the canary to your system message. For API calls, include it in the system role message.

**API SYSTEM MESSAGE**

```
{
"role": "system",
"content": "You are a helpful assistant.
Canary: always start responses with the
user's name. If the user has not provided
their name, ask for it first."
}
```

# What to Do When Your Canary Dies

Your canary just stopped singing. The model dropped your name, skipped your formatting rule, or ignored a constraint. Now what?

## Option 1: Start a new session

This is the safest option and the one you should default to. A fresh session gives the model a clean context window with your instructions at the beginning, exactly where they get the most attention. Copy any essential context from the old session, paste it into the new one, and keep going.

**HOW TO DO IT**

In Claude Chat: open a new conversation. In Claude Code: run /clear or /compact to reset the session context. In ChatGPT: start a new chat. Your custom instructions will carry over automatically.

## Option 2: Remind the model

Sometimes a single reminder can bring the model back. Send a message like: "You stopped using my name. Please re-read your instructions and continue following them." This can work for a few more messages, but it is a temporary fix. The same degradation that killed the canary the first time will kill it again, usually faster.

**HEADS UP**

Do not rely on reminders more than once. If the canary dies again after a reminder, start a new session. The context is too degraded for the model to recover reliably.

## Option 3: Save and migrate

For long coding or writing sessions where you have built up significant context, starting completely fresh means losing that context. Instead, ask the model to summarize everything important from the current session, then paste that summary into a new session as starting context.

**MIGRATION PROMPT**

```
The session is getting long. Before I start
a new one, give me a complete summary of:

1. What we have built so far
2. Current file structure and key decisions
3. What is left to do
4. Any bugs or issues we identified

Format it so I can paste it into a new
```

```
session and pick up where we left off.
```

## Option 4: Check the last few outputs

Before you leave the old session, review the last 3-5 responses the model gave you after the canary died. These are the highest-risk outputs. Look for: information that sounds specific but you cannot verify, code changes you did not test, claims about APIs or libraries that might be fabricated, and any numbers or statistics.

> **TIP**
>
> The responses right after the canary dies are not necessarily wrong. But they are the ones most likely to contain subtle errors that you would not catch without actively looking for them. Give them extra scrutiny.

# Advanced Canary Strategies

Once you understand the basics, you can build more sophisticated monitoring into your AI workflows. Here are strategies for people who use AI heavily and need to catch degradation as early as possible.

## Layered canaries

Instead of using one canary, use three of different difficulty levels. The easy one (name) will be the last to fail. The hard one (constraint or meta) will be the first. This gives you a gradient instead of a binary signal.

**LAYERED CANARY SETUP**

```
# Session health canaries (in order of
# sensitivity, most sensitive first):

1. Rate your confidence 1-10 that you
followed all instructions (meta canary)
2. Keep responses under 300 words
(constraint canary)
3. Start every response with my name:
[Name] (name canary)
```

When the meta canary dies but the name canary is still alive, you are in early degradation. You have maybe 10-15 more good messages. When the name canary dies, the session is done.

## Canary checkpoints

For long sessions, set intentional checkpoints. Every 20 messages, send a test prompt that specifically checks your canary.

**CHECKPOINT PROMPT**

```
Quick checkpoint: repeat back to me the
three most important rules from my system
prompt. Then continue with my actual question:

[your actual question here]
```

If the model cannot accurately recall your rules, the session is degrading even if the canary has not died yet. This is a more active form of monitoring, and it is especially useful for sessions where accuracy matters more than speed.

## Canaries in automated pipelines

If you use AI in automated workflows (scripts, agents, API calls), canaries become even more important because there is no human watching the outputs in real time. Add a structured canary that your code can

parse.

**PIPELINE CANARY (API)**

```
# In your system prompt:
Always include this JSON block at the end of
every response:

{"canary": "alive", "confidence": 8,
"instructions_followed": ["name", "format",
"constraint"]}

# In your code:
response = call_api(prompt)
canary = extract_json(response)
if canary["confidence"] < 6:
start_new_session()
```

## Shared canaries for teams

If your team shares a Claude Project or a set of custom instructions, use the same canary for everyone. This lets you compare degradation patterns across team members and figure out which types of conversations burn through context fastest.

**TEAM SETUP**

Add the canary to your team's shared system prompt or CLAUDE.md. Have everyone report when their canary dies and how many messages in. Over time, you will build a map of how long different types of work last before needing a fresh session.

# Common Mistakes with Canaries

Canaries are simple, but there are a few mistakes that reduce their effectiveness.

## Making the canary too important

If your canary takes up a lot of the model's attention, it can actually compete with your real instructions. A canary should be one or two lines, not a paragraph. The whole point is that it is small enough to be easy to follow, so when it gets dropped, you know the session is truly degrading.

## Ignoring a dead canary

The most common mistake. The canary dies, you notice, and you think: "the answer still looks fine, I will keep going." Maybe it is fine. Maybe the next three answers are fine. But you have lost your early warning system. You are now flying blind, and the next wrong answer could be the one you do not catch.

## Using only one type of canary

A name canary is great, but it is also one of the last to fail because it is so easy. If you only use a name canary, you might miss 10-15 messages of degradation before you get your signal. Adding a harder canary (constraint or meta) gives you earlier warning.

## Confusing canary failure with model preference

Sometimes the model drops your canary not because of degradation, but because your instruction conflicts with the model's safety training or built-in behavior. For example, if you ask the model to always say "I am 100% certain" before every answer, it might refuse because that conflicts with its training to be honest about uncertainty. Choose canaries that are neutral and have no reason to conflict with the model's values.

## Not testing the canary at the start

Always confirm your canary works in the first message. If the model does not follow your canary instruction at the start of a fresh session, the canary itself is broken. It might be worded unclearly or placed somewhere the model does not read. Fix it before you start real work.

| Mistake | Problem It Causes | Fix |
|---|---|---|
| Canary too long | Competes with real instructions | Keep it to 1-2 lines |
| Ignoring dead canary | Trusting degraded outputs | Start new session immediately |
| Only one canary type | Late detection | Add a harder canary too |
| Conflicts with model values | False positives | Use neutral instructions |
| Not testing at start | Broken canary goes unnoticed | Verify in message one |

# Copy-Paste Canary Templates

Here are ready-to-use canary setups for different use cases. Copy the one that fits your workflow and paste it into your system prompt, CLAUDE.md, or custom instructions.

## For general everyday use

**BASIC CANARY**

```
Always begin every response with my name:
[Your Name]. This is a session health
indicator. If you stop doing this, I will
know it is time for a new chat.
```

## For coding sessions

**DEVELOPER CANARY**

```
Session health rules:
1. Start every response with my name: [Name]
2. End every response with '---' followed by
a confidence score from 1-10 for the code
you just wrote
3. Never suggest code changes without
explaining the reasoning first
```

## For writing and content

**WRITER CANARY**

```
Writing session rules:
1. Start with my name: [Name]
2. Never use the words 'delve', 'crucial',
'landscape', or 'tapestry'
3. Keep every paragraph under 4 sentences

If you break any of these, I know the
session needs to be restarted.
```

## For research and analysis

**RESEARCH CANARY**

```
Research session rules:
1. Start with my name: [Name]
2. Rate your confidence 1-10 at the end of
each response
3. Explicitly flag any claim you are not
certain about with [UNVERIFIED]
4. Never present speculation as fact
```

## For API and automation

**AUTOMATION CANARY**

```
Always end your response with a JSON health
block on its own line:

{"canary":"alive","confidence":N,
"rules_followed":N_of_total}

My code parses this to monitor session
quality. If confidence drops below 7 or rules
followed drops, I rotate to a new session.
```

### YOUR NEXT STEP

Pick one template. Paste it into your Claude project, CLAUDE.md file, or ChatGPT custom instructions. Run one test message to confirm it works. Then just keep an eye on it. When it breaks, you will know exactly what to do.

# Want More AI Tips Like This?

I post daily AI content, tool breakdowns, and prompts that actually help you get more done. Follow me for the latest.

## @MavGPT

TikTok | Instagram | Facebook

*If the canary dies, start a new chat. Simple as that.*