

The secret to getting perfect answers from Gemini

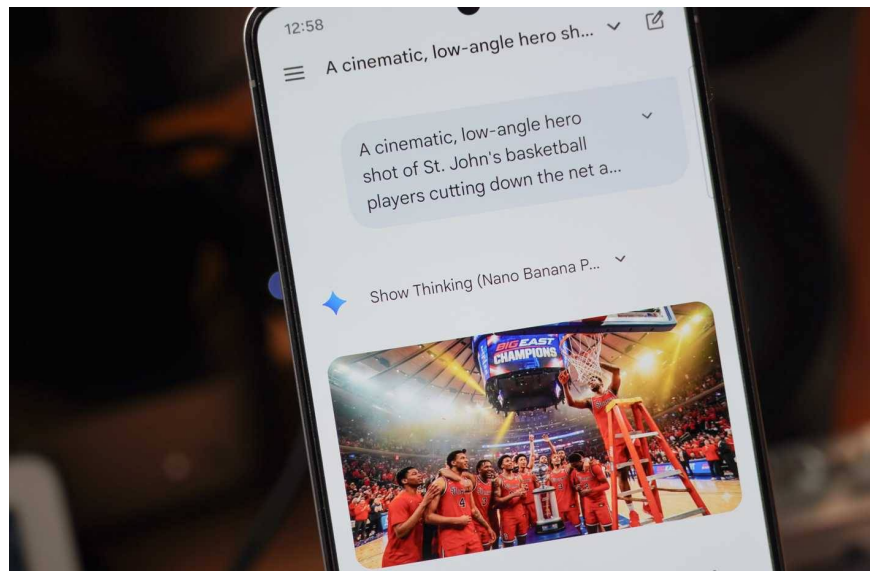
Some long-time Gemini users I've recently learned about a statement generation technique called meta-prompting.

By now, you've probably written quite a few commands for AI chatbots like Gemini or ChatGPT . With powerful large language models (LLMs) supporting these chatbots, Gemini or ChatGPT don't need much instruction to produce good results. However, the quality of the commands you provide to the AI ??model directly correlates with the quality of the results you get. If you want to get better responses from your AI chatbot, you need to provide them with better commands – that's the science.

To improve the quality of your statements, you need to start with statement generation techniques, including optimizing and creating task-specific LLM inputs. Some long-time Gemini users have recently learned about a statement generation technique called meta-prompting. Essentially, this method uses AI to generate the final statements that will be fed into the chatbot. It's an easy way to improve your statement generation skills with virtually no manual effort, and if you haven't used it yet, you're falling behind.

What is meta-prompting?

General guidelines are used to create specific prompts.



If you're new to the world of prompt engineering, this Prompt Engineering Guide is a valuable resource as you learn the different strategies for writing successful AI input prompts. Let's start by explaining what meta-prompting actually is – it's a prompt-building technique that helps AI chatbots like Gemini understand how to approach a problem. Meta prompts add structure and syntax to guide the LLM powering the chatbot in the right direction.

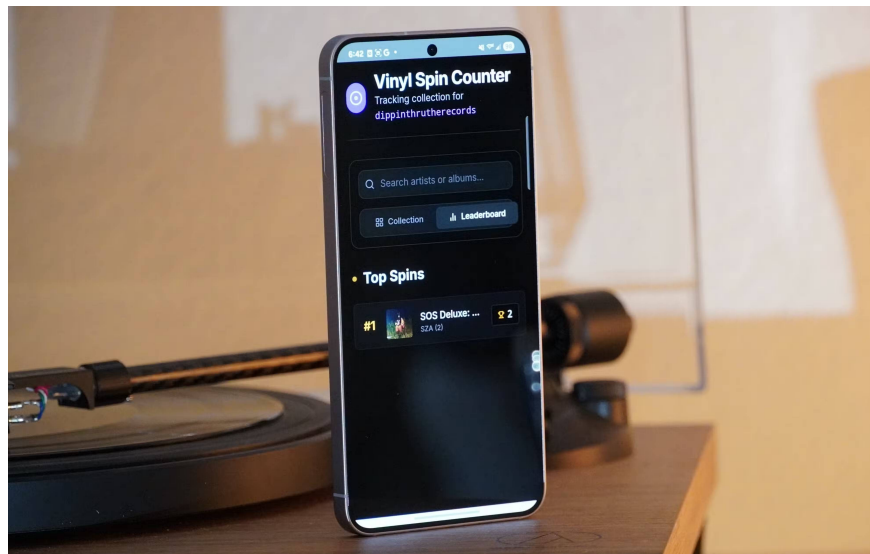
You can write your own meta prompts, but a quicker, easier way to get started with meta-prompting is to have Gemini generate detailed inputs. This idea came from Anna Bortsova, a UX engineer at Google DeepMind, who uses Gemini to create super-level suggestions that are fed into Google's AI tools like Veo 3. Bortsova's suggestions, created with Gemini, can be multi-page long, providing more detail and specificity than human-written guides.

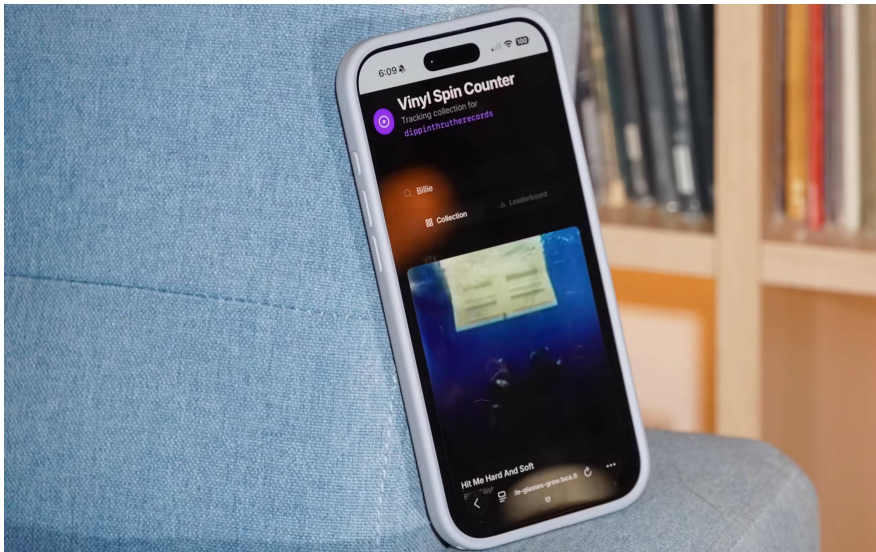
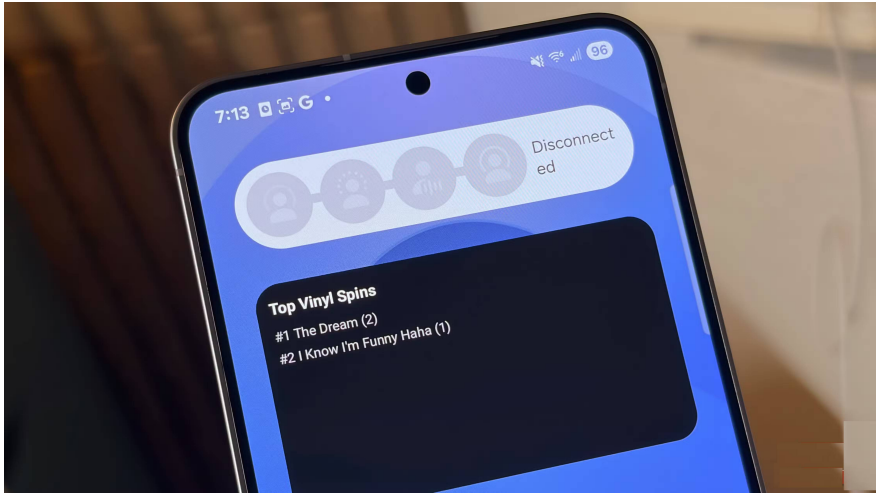
"There are no rules here—we're experimenting—but I've found a few things that help Gemini generate really rich hints," Bortsova writes. "You need to define a very specific task: 'Write a detailed hint that an LLM specialist will understand.' And you need to be clear about your format and style. For example, an 8-second stop-motion animation of scenes rendered on paper. Then, introduce bindings, such as foil or glossy paper, instead of just plain paper. Then, let it do its own thing."

If anyone knows the best way to use Gemini, it's the engineers at Google DeepMind. So, use Gemini to create your own meta-prompting, and the results will be amazing.

Meta-prompting has changed the way people use AI.

Gemini helps users create images and build custom applications.







Meta-prompting with Gemini expands the possibilities for using AI applications, because detailed and optimized prompts become the standard. When using Google's Antigravity application to build custom Android and web applications like the one above, many people didn't write the prompt given to Antigravity. Instead, they gave Gemini the following prompt, asking it to write instructions for an agent-based integrated development environment (IDE):

Create a detailed prompt for an AI-based IDE that will build a simple play counter application for each song in this user's Discogs library: <https://www.discogs.com/user/dippinthrutherecords>

I need a simple interface in the web application so I can count how many times I've played each release and see the rankings of the ones I've played the most.

This only includes albums that I have in my Discogs collection.

Gemini returned a detailed prompt that was somewhat confusing. The meta-prompt generated by Gemini was extremely specific, defining the structure and syntax that Antigravity needed to use to respond to the user's request. It included in-depth programming knowledge to guide the AI agent in the right direction. All of this came from the very simple prompt you see above, and the result was amazing.

This is the meta prompt Gemini provided for Antigravity:

Role

You are a professional React/Next.js developer with experience integrating third-party APIs and managing local state.

Project Objectives

Build a personal web application called "Vinyl Spin Counter".

The application needs to retrieve a specific Discogs user's disc collection and provide a simple interface to track the number of times each disc has been played.

Data Source

** **Target users:** `dippinthrutherecords`*

** **Source URL:** <https://www.discogs.com/user/dippinthrutherecords>*

***API:** Use the Discogs API (`GET /users/{username}/collection/folders/0/releases``) to retrieve the collection.*

**Note:* Focus only on the standard "Collection" (folder ID 0). Do not retrieve the desired list.*

Technology used

** **Framework:** Next.js (App Router).*

** **Style:** Tailwind CSS.*

** **Icon:** Lucide React (for the play button/rating icon).*

** **Data Storage:** Use `localStorage`` for MVP (Minimum Viable Product) to store play counts, ensuring data is persisted between page reloads.*

** **State Management:** React Hooks (Zustand or the standard `useState`/useEffect``).*

Key Features

1. Collection Grid (Home View)

** Retrieve all releases from the `dippinthrutherecords`` collection.*

** Display them in a responsive grid (using Tailwind).*

** **Card Layout:***

** Album cover image (Thumbnail).*

** Artist Name & Album Title (Abbreviate if too long).*

** **Distinctive Counter User Interface:** A large number displays "Total Plays" with a large `+`` button to increase the number of plays and a smaller `-`` button to correct errors.*

2. Leaderboard (Statistics View)

** A separate tab or pop-up window displays a ranking list of "Top 10 most played songs".*

** Sort the data based on the highest number of plays stored locally.*

3. Search & Filter

** A simple text search bar at the top allows you to filter the grid by Artist Name or Album Name instantly.*

Step-by-step implementation plan

- 1. **Setup & Data Types:** Initialize the project and define the TypeScript interface for `DiscogsRelease`.*
- 2. **Client API:** Create a utility function to retrieve collection data. **Important:** Handle pagination if the collection is large (Discogs paginates the results).**
- 3. **State Logic:** Create a custom hook `useSpinTracker` to:*
 - * Load the number of plays from `localStorage` when mounted.*
 - * Combine the number of plays with the retrieved Discogs data (using the release ID as the key).*
 - * Display the functions `incrementSpin(id)` and `decrementSpin(id)`.*
- 4. **User Interface Components:** Build the `ReleaseCard` and `Leaderboard` components.*
- 5. **Assemble:** Assemble the main page.*

Special Instructions

** **API Token:** If the Discogs API requires authentication for this public collection, allow me to define `DISCOGS_TOKEN` in the `.env.local` file and use it in the header.*

** **Aesthetics:** Dark mode by default. Uses the "Vinyl Record" style (dark gray, circular pattern for the profile picture).*

Let's start by setting up the project and defining the API data retrieval logic.

With just a few additional prompts, Antigravity was able to create a custom Vinyl Spin Counter application based on Gemini's instructions. The working and installable Android APK application demonstrated all the benefits of AI-powered prompting. The end result was better than simply providing Antigravity with a plain, human-written prompt. Furthermore, the AI-powered prompting strategy was more efficient, requiring fewer requests. This helped professional users avoid exceeding rate limits or wasting tokens too quickly.

While you can use AI chatbots to help write AI-powered prompts for complex tasks, they also work well for simpler ones. In the image gallery below, you can see the difference between images created using custom prompts and those created using Gemini's meta prompts:

generate an image of the 2024-25 st. johns basketball team winning the big east tournament

Show thinking (Nano Banana Pro)



Share icons: share, copy, refresh, back, menu

Ask Gemini 3

+ Tools

Pro 

Gemini can make mistakes, so double-check it

Here are a few detailed prompt options optimized for high-quality image generators. I have designed these to include specific visual anchors—like the colors, the arena architecture, and the atmosphere—to ensure the best results.

Since "Nano Banana" likely functions similarly to other high-end generative tools (like Midjourney or Flux), I have focused on **lighting, composition, and texture keywords**.

Option 1: The "Photorealistic Broadcast" Prompt (Best for Realism)

This prompt focuses on making the image look like a high-resolution photograph taken by a sports photographer at the exact moment of victory.

Prompt: A hyper-realistic, wide-angle sports photography shot of the 2024-2025 St. John's Red Storm men's basketball team celebrating on the court at Madison Square Garden after winning the Big East Tournament Championship. Central focus on players in red and white "St. John's" uniforms screaming in joy and hoisting the large silver Big East trophy. A blizzard of red and white confetti is falling. The background features the iconic concave ceiling of MSG and a blurred, packed crowd going wild. Dramatic arena lighting, sweat textures on skin, 8k resolution, cinematic depth of field, Nikon Z9 style, high shutter speed capture.

Ask Gemini 3

+ Tools

Pro 

Gemini can make mistakes, so double-check it

Option 2: The "Heroic Portrait" Prompt (Best for artistic style)

This prompt creates a more dramatic, movie-poster style image with a focus on a few key players and the coach.

Prompt: A cinematic, low-angle hero shot of St. John's basketball players cutting down the net after a Big East Championship victory. Dynamic lighting highlighting the "St. John's" text on bright red jerseys. The players are emotional, hugging, with the silver trophy visible in the foreground. In the background, the scoreboard shows "CHAMPIONS". Atmospheric haze, volumetric lighting, vibrant red color palette, sharp focus, unreal engine 5 render style, triumphant atmosphere.

Tips for Optimizing Your Results

If you want to tweak the image, here is a breakdown of why I included specific keywords in the prompts above. You can add or remove these to adjust the output:

- **"Madison Square Garden / Iconic Ceiling":** The Big East Tournament is famously played at MSG. Adding this ensures the background looks like the correct location rather than a generic gym.
- **"Red and White Confetti":** This helps reinforce the team color palette across the entire image, not just on the jerseys.




Ask Gemini 3






+  Tools Pro  

Gemini can make mistakes, so double-check it

Hi Brady
Fast is now powered by 3 Flash. Try it [→](#)

A hyper-realistic, wide-angle sports photography shot of the 2024-2025 St. John's Red Storm men's basketball team celebrating on the court at Madison Square Garden after winning the Big East Tournament Championship. Central focus on players in red and white "St. John's" uniforms screaming in joy and hoisting the large silver Big East trophy. A blizzard of red and white confetti is falling. The background features the iconic concave ceiling of MSG and a blurred, packed crowd going wild. Dramatic arena lighting, sweat textures on skin, 8k resolution, cinematic depth of field, Nikon Z9 style, high shutter speed capture.

+  Tools Pro  

-  Create image
-  Create video
-  Write anything
-  Help me learn
-  Boost my day
-  Find inspiration

Gemini

A hyper-realistic, wide-angle sports photography shot of the 2024-2025 St. John's Red Storm men's basketball team celebrating on the court at Madison Square Garden after winning the Big East Tournament Championship. Central focus on players in red and whit...



A hyper-realistic, wide-angle sports photography shot of the 2024-2025 St. John's Red Storm men's basketball team celebrating on the court at Madison Square Garden after winning the Big East Tournament Championship. Central focus on players in red and whit...

Show thinking (Nano Banana Pro)



Ask Gemini 3

+ Tools

Pro

Gemini can make mistakes, so double-check it

Gemini

A cinematic, low-angle hero shot of St. John's basketball players cutting do...



A cinematic, low-angle hero shot of St. John's basketball players cutting down the net after a Big East Championship victory. Dynamic lighting highlighting the "St. John's" text on bright red jerseys. The players are emotional, hugging, with the silver trophy visible in the...

Show thinking (Nano Banana Pro)



Ask Gemini 3

+ Tools

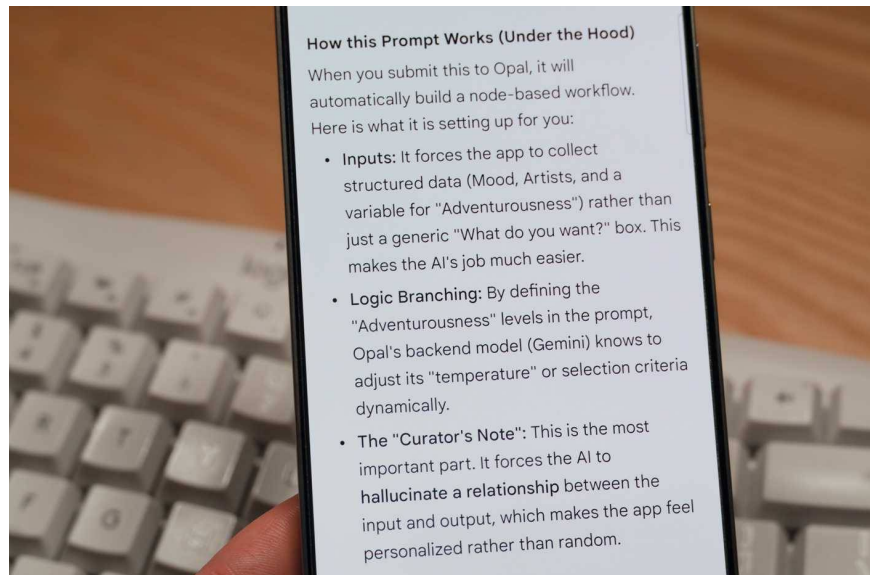
Pro

Gemini can make mistakes, so double-check it

There are very few instances where you wouldn't want to use an AI chatbot to write desired prompts on your behalf. There are virtually no downsides – it's often faster than meticulously writing prompts yourself, and including additional details will improve the quality of Gemini's output.

When should you use meta-prompting?

Whenever specificity is important, use AI to write suggestive questions.



By now, you're probably wondering when to use Gemini to generate meta prompts. Meta prompting can be useful for any LLM-related task, but it's most helpful for tasks and topics you're unfamiliar with. For example, when using Gemini to write a programming prompt for Google Antigravity, the meta prompt it generates includes computer science knowledge you would never have encountered.

Whether you're trying to create an image or build an application, meta prompting with Gemini can help you achieve better results.

You finished reading the article "**The secret to getting perfect answers from Gemini**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.