

[Open in app](#)



Search

Write



Generative AI

Member-only story

Google Nano Banana [Full Guide] Gemini 2.5 Flash API and 100+ Nano Banana Prompts



Simranjeet Singh

[Follow](#)

13 min read · Sep 24, 2025

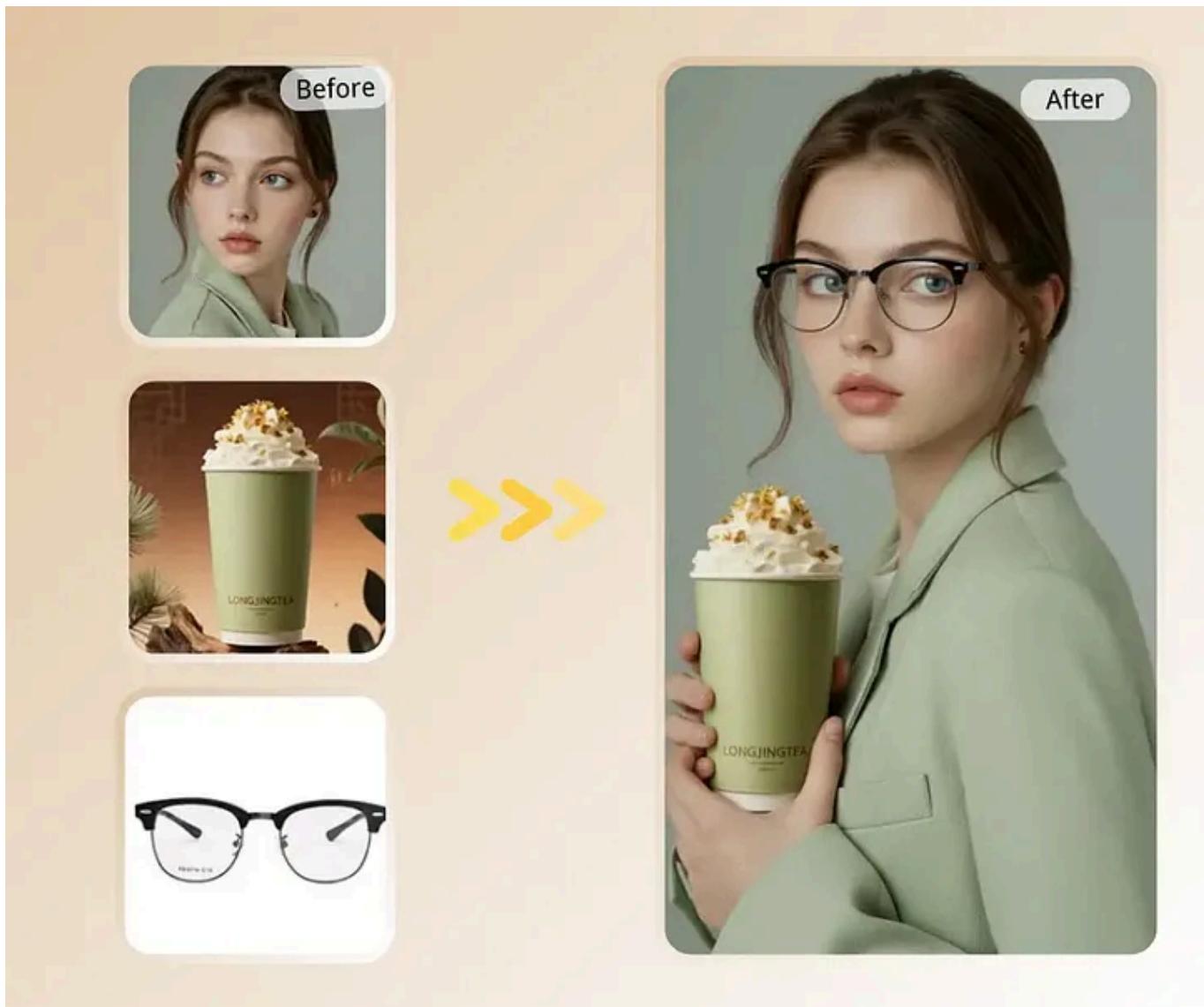
248

5



...

“This isn’t just a filter — this is Nano-Banana, where your photos become magic with Python (yes, you can code it).”



Google Nano Banana—Image Synthesis [Multiplying Images in One]

Nano-Banana is Google's Gemini 2.5 Flash Image AI, a **next-gen image editing and generation model**. It lets users turn a single photo into multiple creative outputs, blend multiple images, and perform style transfers with remarkable precision.

Why it's trending:

- Viral image edits on social media.
- Character-consistent transformations.

- Multi-image fusion for surreal, high-quality outputs.

20+ Prompts



Full Guide

Python Implementation



Google provides an API that allows you to integrate Nano-Banana into Python scripts, enabling automated image editing, batch transformations, and prompt-driven generation.

Goal of the Blog

1. How does Nano-Banana work?
2. Technical Details
3. Python API usage
4. Tips for Crafting Prompts
5. 100+ Trending example prompts.

Example Prompt Ideas:

- “Turn this selfie into a 90s grunge portrait with a soft cinematic glow.”
- “Blend these two photos: a mountain sunset and a city skyline at night.”
- “Change the hairstyle to a futuristic neon look while keeping the original face intact.”

How does Nano-Banana work?

“From Gemini’s lab to your studio — what powers Nano-Banana”

Google’s Gemini 2.5 Flash Image, colloquially known as Nano-Banana, is an advanced AI model designed for high-quality image generation and editing. It enables users to create and modify images using natural language prompts, offering unprecedented control over visual content.

Key Features:

- **Image Generation & Editing:** Create and edit images based on textual descriptions.
- **Multi-Image Fusion:** Combine multiple images into a single cohesive output.
- **Character Consistency:** Maintain consistent characters across different images for storytelling.
- **Targeted Edits:** Make specific alterations to images, such as changing styles or adding elements.

Pricing / Token Usage:

- **Standard Tier:** \$30 per 1 million output tokens.
- **Image Cost:** Each image consumes approximately 1,290 tokens, equating to about \$0.039 per image.
- **Input Pricing:** Text or image inputs are priced at \$0.30 per million tokens

Technical Details

“You like the output. Here’s what’s happening behind the scenes.”

Google’s “Nano Banana” is a vision-language model (VLM) that stands out for its speed, efficiency, and ability to understand and manipulate images with natural language commands. Unlike some older models, it fuses text and image processing in a single, unified architecture.

How Vision-Language Models (VLMs) Work

Vision-language models are a type of **multimodal AI** that can process and understand both visual (images, videos) and textual data simultaneously. At a high level, a VLM works by combining three key components:

- **Vision Encoder:** This component processes an image and converts it into a numerical representation called an **embedding**. It breaks the image down into smaller patches, similar to how a language model tokenizes text, and uses a **Vision Transformer (ViT)** to understand the relationships between these patches.
- **Language Encoder:** This component, often a standard **transformer-based large language model (LLM)**, processes the text prompt. It tokenizes the input words and converts them into numerical embeddings, capturing their meaning and context.
- **Fusion Mechanism:** This is the core of the VLM. It's a crucial part of the architecture that aligns the image embeddings from the vision encoder and the text embeddings from the language encoder into a shared "language." This allows the model to understand how the visual and textual information relate to each other. For example, it learns to associate the image of a "cat" with the text "cat" and to disassociate it from the text "dog".

Once trained, this fused representation allows the model to perform tasks like image captioning, visual question answering, or, in the case of Nano Banana, image generation and editing. When a user provides a prompt like "**Making 3d-Figures**", the model understands both the visual data (the image of the red car) and the linguistic instruction (change to blue) to generate the desired output.



Rohit Sharma and Family in 3D Figure

Nano Banana's Technical Details 🍌

Nano Banana's underlying architecture is an advanced VLM built on Google's **Gemini architecture**. It is designed for efficiency and speed, with sub-2-second inference times for image generation. Here are some of its key technical features:

- **Unified Multimodal Architecture:** Instead of relying on separate “translation layers”, Nano Banana processes text prompts and visual data together in a **single transformer system**. This native fusion allows for better semantic understanding and more precise edits that maintain the context and realism of the original image.
- **Diffusion-based Generation:** The model uses **diffusion techniques** for image generation. Diffusion models start with a noisy image and progressively “denoise” it based on the input prompt to create a coherent final image. Nano Banana integrates this with its Gemini architecture for enhanced capabilities.

- **On-Device and Cloud Optimization:** While Nano Banana requires significant cloud resources for its full capabilities, its core technology is optimized for on-device processing. This is a key feature of the **Gemini Nano** model, which can run on devices like the Pixel phone, enabling offline features like image description and summarization.
- **Model Distillation:** To achieve its speed and efficiency, Nano Banana uses **model distillation**, a process where a smaller, more efficient model learns from the knowledge of a larger, more complex one. This allows it to perform complex tasks with fewer computational resources.
- **Context Window:** The model can work with a massive 1-million token context window, which is especially useful for long-range reasoning and for editing multiple images while maintaining a consistent style or subject identity.
- **Multimodal Reasoning:** Nano Banana can take multiple input images and synthesize them into a single cohesive output while respecting physical properties like light and shadows. This enables complex tasks like seamlessly blending a product photo into an interior shot.
- **SynthID Watermarking:** All images created or edited with the model are embedded with an **invisible SynthID digital watermark**. This helps to identify them as AI-generated, promoting trust and safety.



Google Nano Banana SynthID Watermarking [Bottom Right]

Input Types:

Nano Banana supports various input combinations:

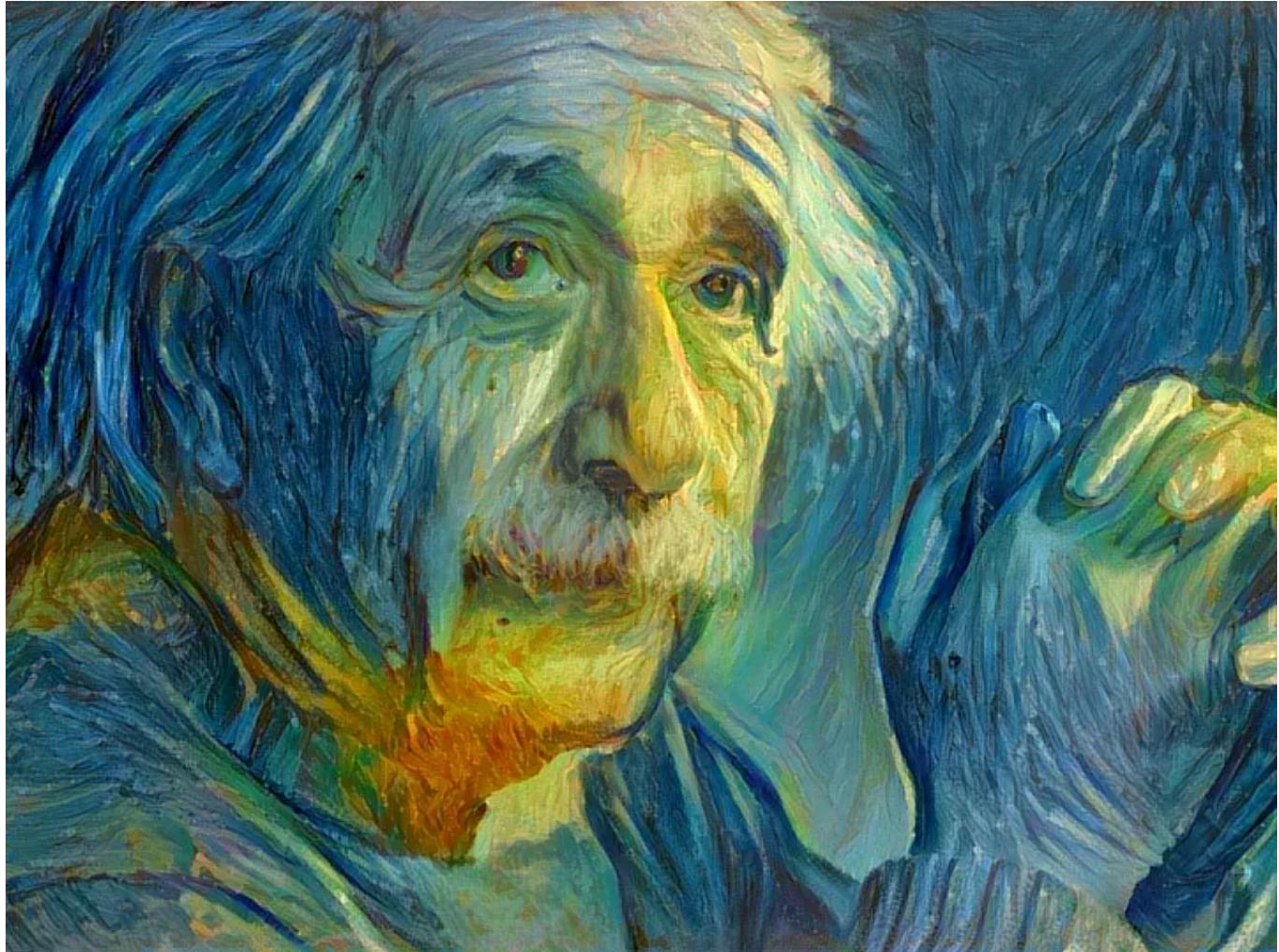
- **Text Prompt Only:** Generate images from descriptive text.
- **Image + Text for Editing:** Modify existing images based on textual instructions.
- **Multiple Images for Fusion:** Combine up to five images to create a new composition.

Example:

Input: A photo of a beach at sunset.

Prompt: “Transform this into a painting in the style of Van Gogh.”

Output: An impressionistic painting of the beach scene.



“Transform this into a painting in the style of Van Gogh.”

Output Format & Consistency:

Nano Banana ensures:

- **Character Identity Maintenance:** Consistent appearance of characters across different images.
- **Style Retention:** Preservation of artistic styles and themes.

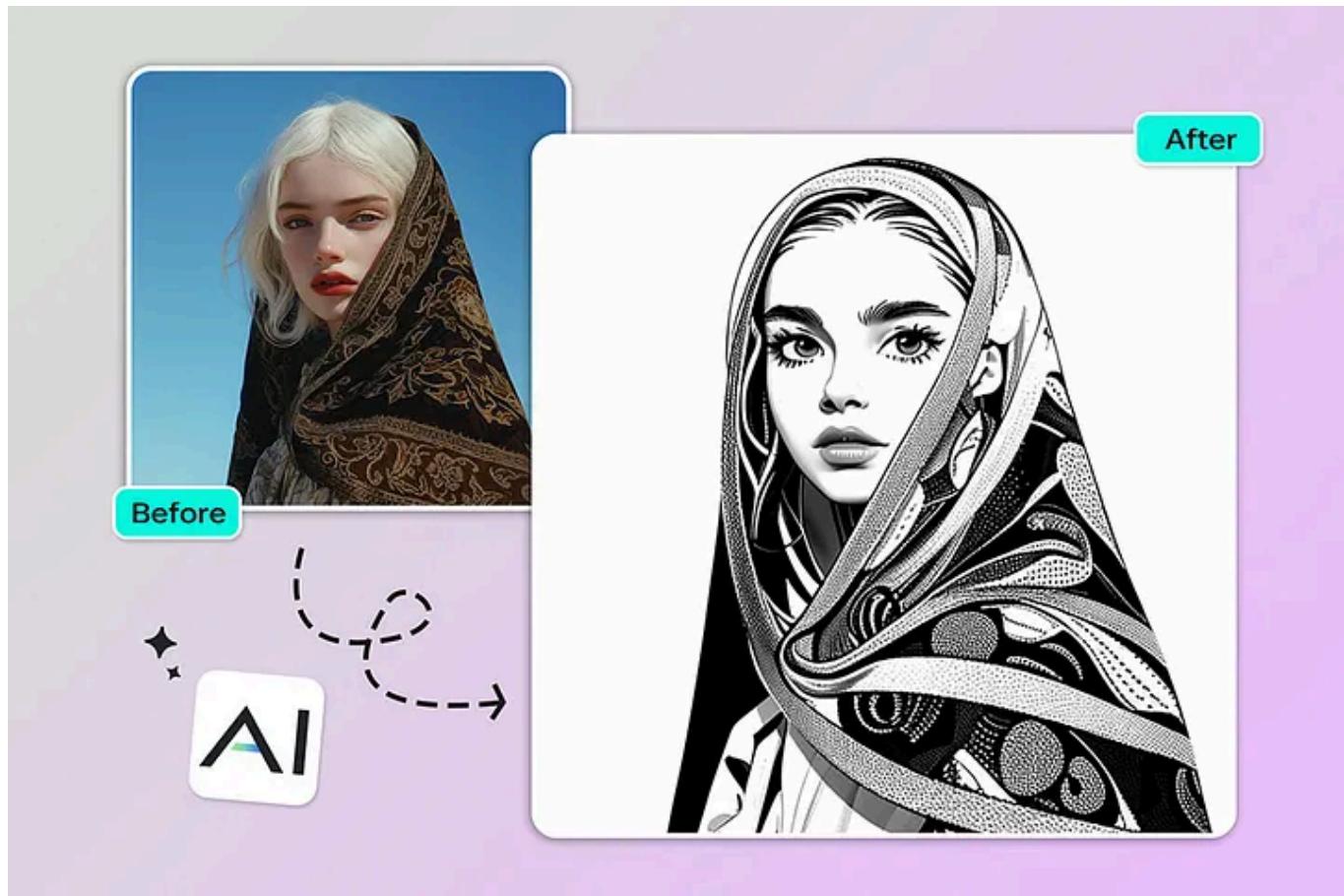
- **Realism vs. Stylisation Balance:** Ability to generate both photorealistic and stylised images based on prompts.

Example:

Input: A portrait of a person.

Prompt: “Create a comic book version of this portrait.”

Output: A stylized comic book illustration of the person.



“Create a comic book version of this portrait.”

Prompt Processing:

Effective prompts should include:

- **Detail Level:** Specific descriptions yield more accurate results.
- **Style Indicators:** Mentioning artistic styles guides the model's output.
- **Aspect Ratios:** Specifying dimensions ensures the image fits desired formats.

Example:

Prompt: “A futuristic cityscape at night, in cyberpunk style, aspect ratio 16:9.”

Output: A cyberpunk-themed cityscape image with the specified aspect ratio.



“A futuristic cityscape at night, in cyberpunk style, aspect ratio 16:9.”

Tokenisation & Cost:

Pricing for Nano Banana is based on token usage:

- **Output Pricing:** \$30 per 1 million output tokens.
- **Image Cost:** Approximately 1,290 tokens per image, equating to about \$0.039 per image.

Example:

Scenario: Generating 100 images would cost approximately \$3.90.

Example Prompt Workflow:

- 1. Input:** A photo of a Girl.
- 2. Prompt:** “Transform this photo into a cartoon character in the style of Pixar.”
- 3. Output:** A cartoon version of the Girl resembling Pixar animation.



“Transform this photo into a cartoon character in the style of Pixar.”

Setup & Python API Integration

While end-users can explore the magic of Nano Banana directly in the Gemini app, developers looking to **prototype, experiment, and perfect AI prompts** will find their ultimate playground in Google AI Studio. Think of AI Studio as your creative lab — a space where you can **test every AI model available, refine your ideas, and prepare for coding without limits**. It's also your gateway to building powerful applications with the **Gemini API**.

The best part? You can **use Nano Banana completely free** within AI Studio. Getting started is a breeze:

Visit aistudio.google.com, sign in with your Google account, and select **Nano Banana** from the model picker.

Ready to jump in?

Start a new session instantly with **direct access to Nano Banana here: ai.studio/banana**.



```
prompt = "Restore and colorize this image from 1932"
```

```
response = client.models.generate_content(  
    model="gemini-2.5-flash-image-preview",  
    contents=[prompt, image],  
)
```

The screenshot shows the Google AI Studio interface with a dark theme. On the left, there's a sidebar with options like Studio (Chat, Stream, Generate media, Build, History), Dashboard, Documentation, and a note about the model being unstable. In the center, there's a main area titled "Google AI Studio" with a sub-section for "add a cool cap with a banana in comic style". This section includes a preview image of a man with a beard, a "Run" button, and a note about Gemini 2.5 Flash Image not supporting image editing. To the right, there's a "Run settings" panel for a task named "Nano Banana" using the "gemini-2.5-flash-image-preview" model. It shows token count (1,302 / 32,768), temperature (1), and advanced settings like safety settings, add stop sequence, output length (8192), and top p (0.95). Three pink arrows point from the text below to specific parts of the interface: one arrow points to the "Run settings" panel with the text "1. Select Nano Banana"; another arrow points to the "Create & edit images" section with the text "2. Create & edit images"; and a third arrow points to the "Get API key" button with the text "3. Create API key".

Google AI Studio Interface

To follow this guide, you will need the following:

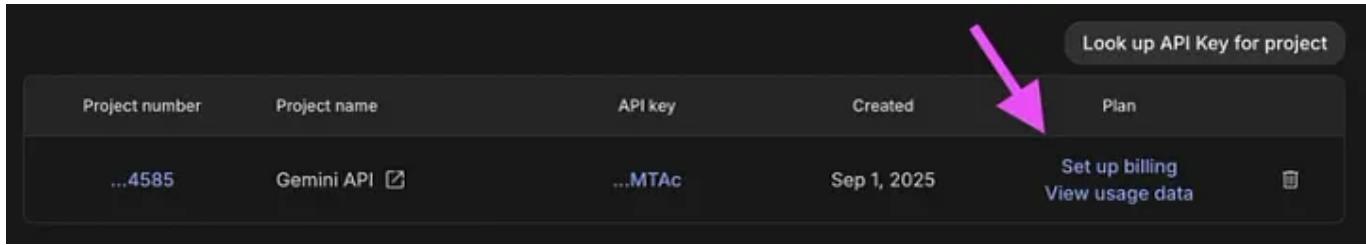
- An API key from [Google AI Studio](#).
- Billing is set up for your project.
- The Google Gen AI SDK for [Python](#) or [JavaScript/TypeScript](#).

Follow these steps:

- In Google AI Studio, click **Get API key** in the left navigation panel.
- On the next page, click **Create API key**.
- Select an existing Google Cloud project or create a new one. This project is used to manage billing for API usage.

Once the process is complete, your API key will be displayed. Copy and store it securely.

While prototyping in AI Studio is free, using the model via the API is a paid service. You must enable billing on your Google Cloud project.



Project number	Project name	API key	Created	Plan
...4585	Gemini API	...MTAc	Sep 1, 2025	Set up billing View usage data

Install the SDK

Choose the SDK for your preferred language.

```
pip install -U google-genai
# Install the Pillow library for image manipulation
pip install Pillow
```

Python API usage

Image Generation from Text

Use Nano Banana to generate one or more images from a descriptive text prompt. Use the model ID `gemini-2.5-flash-image-preview` for all API requests.

```
from google import genai
from PIL import Image
from io import BytesIO

# Configure the client with your API key
client = genai.Client(api_key="YOUR_API_KEY")

prompt = """Create a photorealistic image of an orange cat
with a green eyes, sitting on a couch."""

# Call the API to generate content
response = client.models.generate_content(
    model="gemini-2.5-flash-image-preview",
    contents=prompt,
)

# The response can contain both text and image data.
# Iterate through the parts to find and save the image.
for part in response.candidates[0].content.parts:
    if part.text is not None:
        print(part.text)
    elif part.inline_data is not None:
        image = Image.open(BytesIO(part.inline_data.data))
        image.save("cat.png")
```



Output

The model is multimodal, so the response is structured as a list of `parts` that can contain interleaved text and image data (`inline_data`). The code above iterates through these parts to extract and save the generated image.

Image Editing with Text and Image Inputs

Provide an existing image along with a text prompt to perform edits. The model excels at maintaining character and content consistency from the

input image.

```
from google import genai
from PIL import Image
from io import BytesIO

client = genai.Client(api_key="YOUR_API_KEY")

prompt = """Using the image of the cat, create a photorealistic,
street-level view of the cat walking along a sidewalk in a
New York City neighborhood, with the blurred legs of pedestrians
and yellow cabs passing by in the background."""

image = Image.open("cat.png")

# Pass both the text prompt and the image in the 'contents' list
response = client.models.generate_content(
    model="gemini-2.5-flash-image-preview",
    contents=[prompt, image],
)

for part in response.candidates[0].content.parts:
    if part.text is not None:
        print(part.text)
    elif part.inline_data is not None:
        image = Image.open(BytesIO(part.inline_data.data))
        image.save("cat2.png")
```



Output

Photo restoration with Nano Banana

One of the model's powerful applications is photo restoration. With a simple prompt, it can restore and colorize old photographs with impressive results.

```
from google import genai
from PIL import Image
from io import BytesIO

client = genai.Client(api_key="YOUR_API_KEY")

prompt = "Restore and colorize this image from 1932"

image = Image.open("lunch.jpg") # "Lunch atop a Skyscraper, 1932"

response = client.models.generate_content(
    model="gemini-2.5-flash-image-preview",
    contents=[prompt, image],
)

for part in response.candidates[0].content.parts:
    if part.text is not None:
        print(part.text)
    elif part.inline_data is not None:
```

```
image = Image.open(BytesIO(part.inline_data.data))
image.save("lunch-restored.png")
```



Output

Working with Multiple Input Images

You can provide multiple images as input for more complex editing tasks.

```
from google import genai
from PIL import Image
from io import BytesIO

client = genai.Client(api_key="YOUR_API_KEY")

prompt = "Make the girl wear this t-shirt. Leave the background unchanged."

image1 = Image.open("girl.png")
image2 = Image.open("tshirt.png")

response = client.models.generate_content(
    model="gemini-2.5-flash-image-preview",
    contents=[prompt, image1, image2],
)

for part in response.candidates[0].content.parts:
    if part.text is not None:
        print(part.text)
```

```
elif part.inline_data is not None:  
    image = Image.open(BytesIO(part.inline_data.data))  
    image.save("girl-with-tshirt.png")
```



Output

Conversational Image Editing

For iterative refinement, you can use a `chats` session to maintain context across multiple requests. This allows you to edit images conversationally.

```
from google import genai  
from PIL import Image  
from io import BytesIO  
  
client = genai.Client(api_key="YOUR_API_KEY")  
  
# Create a chat  
chat = client.chats.create(  
    model="gemini-2.5-flash-image-preview"  
)  
  
# Make the first image edit  
response1 = chat.send_message(  
    [  
        "Change the cat to a bengal cat, leave everything else the same",  
        Image.open("cat.png"),  
    ]  
)
```

```
# display / save image...

# Continue chatting and editing
response2 = chat.send_message("The cat should wear a funny party hat")
# display / save image...
```



Output

Tip: If you notice image features starting to degrade or “drift” after many conversational edits, it’s best to start a new session with the latest image and a more detailed, consolidated prompt to maintain high fidelity.

Google Nano Banana Hackathon Kit

[GitHub - google-gemini/nano-banana-hackathon-kit: 🎉 The official starter kit for the Nano Banana...](#)

🎉 The official starter kit for the Nano Banana Hackathon. Clone this repo to get building fast! ...

github.com

Tips for Crafting Prompts

To achieve the best results with Nano Banana, follow these prompting guidelines:

- **Be Hyper-Specific:** The more detail you provide about subjects, colors, lighting, and composition, the more control you have over the output.
- **Provide Context and Intent:** Explain the purpose or desired mood of the image. The model's understanding of context will influence its creative choices.
- **Iterate and Refine:** Don't expect perfection on the first try. Use the model's conversational ability to make incremental changes and refine your image.
- **Use Step-by-Step Instructions:** For complex scenes, break your prompt into a series of clear, sequential instructions.
- **Use Positive Framing:** Instead of negative prompts like "no cars," describe the desired scene positively: "an empty, deserted street with no signs of traffic."
- **Control the Camera:** Use photographic and cinematic terms to direct the composition, such as "wide-angle shot", "macro shot", or "low-angle perspective".

For a deeper dive into best practices, review the official blog post on [prompting best practices](#) and the [prompting guide](#) in the documentation.

100+ Trending example prompts

Explore what the community is building with Nano Banana:

- Shifting camera perspective by @henrydaubrez: [X post](#)
- Few-shot learning for consistent character design by @multimodalart: [X post](#)

- “What does the red arrow see” Google Maps transforms by @tokumin: [X post](#)
- Generating images from stick figure annotations by @yachimat_manga: [X post](#)
- Creating 3D models from still images by @deedydas: [X post](#)
- Generating location-based AR experiences by @bilawalsidhu: [X post](#)
- Converting a 2D map into a 3D graphic by @demishassabis: [X post](#)

This guide has covered the fundamentals of building with Nano Banana aka Gemini 2.5 Flash Image. You’ve learned how to set up your environment, generate and edit images, and apply advanced techniques. Now you’re ready to start incorporating these powerful capabilities into your own projects.

For further reading, check out the official resources:

- [Google AI Studio](#)
- [Gemini API docs](#)
- [Nano Banana Gemini API docs](#)
- [How to prompt Gemini 2.5 Flash Image Generation for the best results](#)
- [Nano Banana docs prompting guide](#).
- [Pixshop app in AI Studio](#)

Read my Google Interview Experience and Joining as AI/ML Engineer

TCS to Google | Manual Tester to AI/ML Engineer | Tier 3 to FAANG | Interview and Learning...

How I go from TCS Manual Tester to Google AI/ML Engineer with GenAI/LLM skillset. Complete Journey and Learning...

generativeai.pub

If my story inspired you, I invite you to **follow my journey** and stay connected:

- Medium – for detailed articles, tutorials, and insights on AI/ML and GenAI.
- LinkedIn – for daily updates, learning tips, and project showcases.
- YouTube – for hands-on tutorials, GenAI experiments, and career advice.

I've also compiled learning resources and guides that helped me at every stage – from Data Science fundamentals to advanced GenAI and LLM projects. Links to these resources are available in the description and posts.

Remember, your dream is closer than you think. Start small, stay consistent, and never give up. The next success story could be yours.

Keep learning. Keep building. Keep believing.

Looking ahead, I'm excited to share you my **75 Hard GenAI Challenge** in which you learn GenAI for Free from Scratch.

 Agentic AI 14+ Projects- <https://www.youtube.com/playlist?list=PLYIE4hvWhsAkn8VzMWbMOxetpaGp-p4k>

 Learn RAG from Scratch – <https://www.youtube.com/playlist?list=PLYIE4hvWhsAKSZVAn5oX1k0oGQ6Mnf1d>

 Complete Source Code of all 75 Day Hard

 GitHub –

https://github.com/simranjeet97/75DayHard_GenAI_LLM_Challenge

 Kaggle Notebook – <https://www.kaggle.com/simranjeetsingh1430>

 Learn GenAI for Free [Free Courses and Study Material with Daily Updates and Learning's Uploaded] Join Telegram  – <https://t.me/genaiwithsimran>

 Exclusive End to End Projects on GenAI or Deep Learning or Machine Learning in a Domain Specific way –

<https://www.youtube.com/@freebirdscREW2023>

You can also schedule a meeting with me here.

Link – <https://topmate.io/simranjeet97>

If you like the article and would like to support me make sure to:

 Clap for the story (100 Claps) and follow me  Simranjeet Singh

 View more content on my Medium Profile

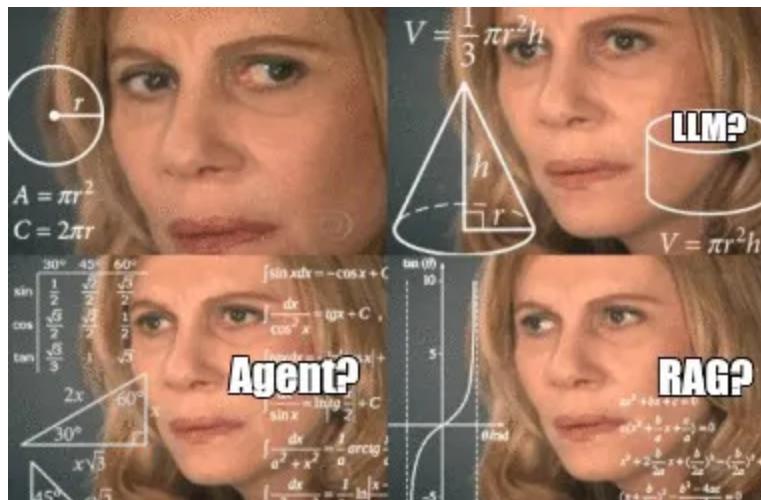
🔔 Follow Me: [LinkedIn](#) | [Medium](#) | [GitHub](#) | [Linkedin](#) | [Telegram](#)

🚀 Help me in reaching to a wider audience by sharing my content with your friends and colleagues.

👉 Do Donate 💰 or Give me a Tip 💸 If you really like my blogs. Click Here to Donate or Tip 💰 – <https://bit.ly/3oTHiz3>

🎓 If you want to start a career in Data Science and Artificial Intelligence you do not know how? I offer data science and AI mentoring sessions and long-term career guidance.

 1:1 Guidance — About Python, Data Science, and Machine Learning



Confused? Schedule call with me on Topmate Now? <https://topmate.io/simranjeet97>

Generative AI

This story is published on [Generative AI](#). Connect with us on [LinkedIn](#) and follow [Zeniteq](#) to stay in the loop with the latest AI stories.

Subscribe to our [newsletter](#) and [YouTube](#) channel to stay updated with the latest news and updates on generative AI. Let's shape the future of AI together!

Google

Google Nano Banana

Artificial Intelligence

Vision Language Model

Technology



Published in Generative AI

60K followers · Last published 4 hours ago

Follow

Stay updated with the latest news, research, and developments in the world of generative AI. We cover everything from AI model updates, comprehensive tutorials, and real-world applications to the broader impact of AI on society. Work with us: jimclydegm@gmail.com



Written by Simranjeet Singh

3.2K followers · 40 following

Follow



AI/ML at GOOGLE | GenAI Expert | Finance and Banking | 3K Medium + 14K YouTube | Machine Learning | Deep Learning | NLP

Responses (5)





Bgerby

What are your thoughts?



Muskan
8 hours ago

...

gemini 2.5 makes nano banana work like a light seo bot. i use it to map topics fast and drop into sheets. it's a clean way to see what google might favor next. runs smooth but rate limits can bite. happy to dm my config file.

👏 2 [Reply](#)



Imran Ali | AI Tools & Gaming Gear Reviews he/him
18 hours ago

...

I couldn't read the full story since it's member-only but the beginning was really well written and hooked me right away

👏 1 [Reply](#)



Scholars Review
1 day ago

...

Insightful

👏 1 [Reply](#)

[See all responses](#)

More from Simranjeet Singh and Generative AI

RAG IS Hard Until I Know these Techniques RAG to 99% Accuracy

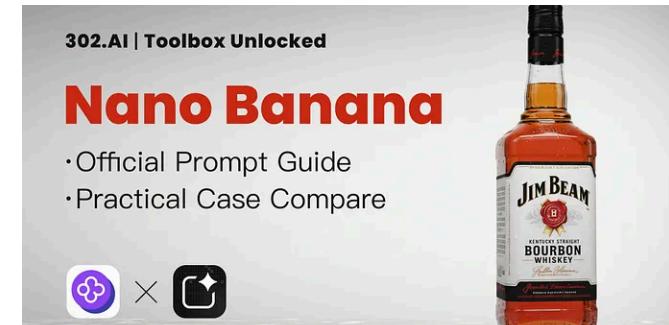


In Artificial Intelligence in Plain ... by Simranjeet S...

RAG is Hard Until I Know these 12 Techniques → RAG Pipeline to 99...

RAG is Hard Until I Know these 12 Techniques
→ RAG Pipeline to 99% Accuracy. Best Blog...

Sep 27 423 6



In Generative AI by 302.AI

Official Nano-Banana Prompting Guide with Test Case Comparisons

Unlock Google's Official Prompt Tips with 6 wild cases to instantly Improve Your Image...

Sep 4 182 5



Hierarchical Reasoning Model (HRM)

**A tiny brain that
embarrasses
giant LLMs**



In Generative AI by Dr. GenAI

Hierarchical Reasoning Model (HRM): a tiny brain that...

HRM is a 27M-parameter model trained in hours that beats giant LLMs on reasoning...



Sep 7



302



6



...

ML Engineer

Interview Questions

Cracking ML Interviews
Years of Dedication



In Artificial Intelligence in Plain ... by Simranjeet S...

OpenAI ML Engineer Interview Questions 2025

A mock interview with an OpenAI ML engineer covering LLM deployment, low-...



Sep 24



136



8

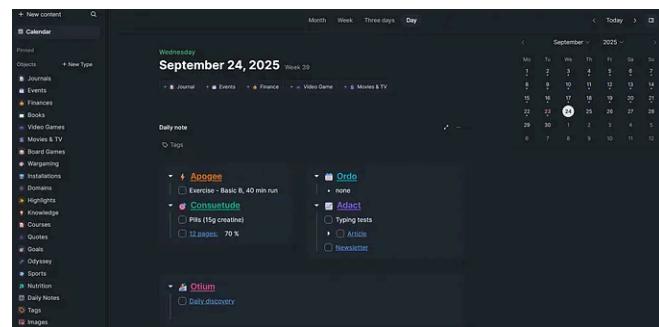


...

See all from Simranjeet Singh

See all from Generative AI

Recommended from Medium





In [AI monks.io](#) by Akram Mubeen

10 Viral Google Nano Banana AI Prompts You Must Try

Nano Banana Prompts That Are Breaking the Internet

⭐ Sep 26 ⚡ 111



♾ In [Level Up Coding](#) by Fareed Khan

Building 17 Agentic AI Patterns and Their Role in Large-Scale AI...

Ensembling, Meta-Control, ToT, Reflexive, PEV and more

⭐ Sep 25 ⚡ 1.8K ⚡ 40



🌐 DJ A

No Censorship AI: The Revolution Will Not Be Sanitized

In this crazy freaking world where even the robots are scared of saying the wrong thing,...

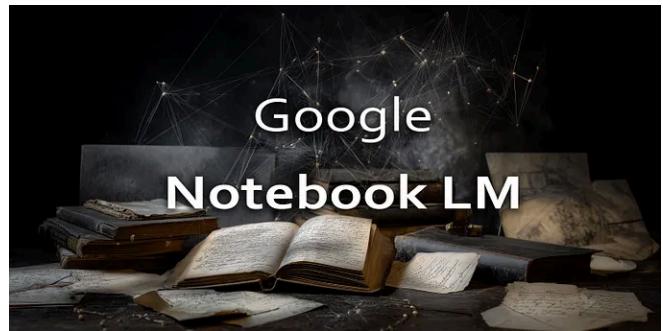


Tosny

7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.

⭐ Sep 23 ⚡ 2.4K ⚡ 84



"urls In [Let There Be Prompt](#) by Aivaras A Grauzinis

I Finally Figured Out How I Can Use Google's NotebookLM

And it might be helpful for you too

⭐ Sep 19 ⚡ 638 ⚡ 13



🌐 Jaytech

11 Free Microsoft AI Tools You Should Know in 2025 🤖

Photo by BolivialInteligente on Unsplash

Sep 30  66  4



...



Oct 1



...

[See more recommendations](#)