**How to Build with Nano Banana: A Complete Developer Guide for Gemini 2.5 Flash Image**

Google has a new tool for us devs: **Gemini 2.5 Flash Image**.

The codename? **Nano Banana.** Yep, that’s what they’re calling it.

Don’t let the silly name fool you — it’s actually one of the strongest image models.

You can do all the usual text-to-image stuff, but it goes further: photo restoration, multiple input images, step-by-step conversational edits, even stitching together wild scenarios from a few quick prompts. Think of it as Photoshop with a bit of magic.

In this post I’ll show you how to get started — from zero setup to generating your first AI-powered cat photo.



Press enter or click to view image in full size

Image is generated by — Google AI studio (Nano Banana**)**

**Getting Started**

The easiest way to test Nano Banana before writing a single line of code is **Google AI Studio**. You just log in with your Google account, pick **Nano Banana** from the model list, and throw some prompts at it.

Want a shortcut? [Here’s the direct link](https://ai.studio/banana).

Pro tip: AI Studio also has a section called **Apps**, where you can remix small projects. Handy if you want to see what others are building.

**Setting Up Your Project**

Alright, if you want to actually build something with code, you’ll need three things:

1. An **API key** from AI Studio
2. Billing turned on in Google Cloud
3. The **Gen AI SDK** (Python or JS)

**Grab an API key**

* Go to AI Studio → **Get API key** → **Create key**
* Pick a Google Cloud project (or spin up a new one)
* Copy your shiny new key and tuck it somewhere safe

**Billing**

Testing in AI Studio is free, but API calls cost a bit. Each generated image comes out to about **$0.039** (so, roughly 25 images for a dollar). Not bad for what it can do.

**Install the SDK**

Python:

pip install -U google-genai  
pip install Pillow

JavaScript:

npm install @google/genai

I’ll stick with Python for the rest of this tutorial, but the JS SDK works just as well.

**Your First Image**

Let’s make a cat. Because why not?

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")

Run that, and boom — you’ve got cat.png sitting in your folder.



Press enter or click to view image in full size

Image is generated by — Google AI studio (Nano Banana**)**

**Editing an Image with a Prompt**

Now let’s take that same cat and move it to New York.

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")

Nano Banana is surprisingly good at keeping the “same” cat, just moving it into a new scene.



Press enter or click to view image in full size

Image is generated by — Google AI studio (Nano Banana**)**

**Restoring Old Photos**

This part impressed me the most. You can throw an old black-and-white photo at it and just say “restore and colorize.”

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")

It doesn’t just slap on random colors — it actually tries to match realistic tones.



Press enter or click to view image in full size

Image is generated by — Google AI studio (Nano Banana**)**

**Using Multiple Images**

Want to merge two photos? Easy.

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")

That’s the kind of workflow that would usually need Photoshop + a decent designer. Here it’s just a function call.



Press enter or click to view image in full size

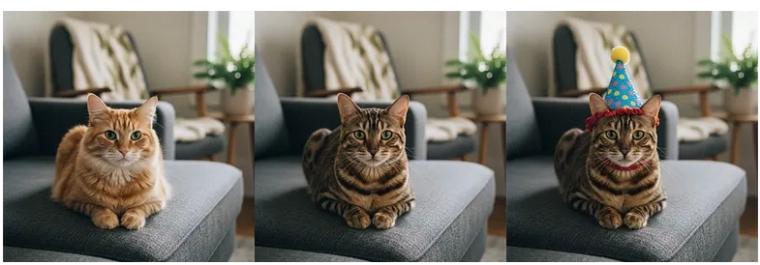
Image is generated by — Google AI studio (Nano Banana**)**

**Conversational Edits**

Nano Banana can remember context if you use chat sessions. This means you don’t have to rewrite your whole prompt each time.

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...

You can keep nudging it step by step, which feels pretty natural.



Press enter or click to view image in full size

Image is generated by — Google AI studio (Nano Banana**)**

**Prompting Tips**

Here’s what worked best for me:

* **Be specific.** Don’t just say “dog.” Say “a golden retriever puppy running on grass, morning sunlight.”
* **Add context.** If you want it for a book cover, mention that — it changes the vibe.
* **Iterate.** First prompt won’t always be perfect. Nudge it along.
* **Stay positive.** Instead of “no cars,” say “an empty street.”
* **Think like a photographer.** Wide-angle, macro, low-light — it gets those terms.

**What Others Are Building**

Some cool stuff I’ve seen floating around:

* Maps turned into 3D graphics
* Stick figure doodles becoming finished art
* Consistent characters across multiple scenes
* Restored family photos shared on socials

It’s early days, but people are already pushing it in interesting directions.