Image generation with Gemini (aka Nano Banana & Nano Banana Pro)

Gemini can generate and process images conversationally. You can use either Gemini 2.5 Flash (aka Nano Banana) or Gemini 3 Pro Preview (aka Nano Banana Pro) with text, images, or a combination of both. This lets you create, edit, and iterate on visuals with unprecedented control.

All generated images include a SynthID watermark (/responsible/docs/safeguards/synthid).

Image generation (text-to-image)

Python (#python) JavaScript (#javascript) Go (#go) Java (#java) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flasl
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "parts": [
        {"text": "Create a picture of a nano banana dish in a fancy restau
      ]
    }]
  }'
```

Image editing (text-and-image-to-image)

Reminder: Make sure you have the necessary rights to any images you upload. Don't generate content that infringe on others' rights, including videos or images that deceive, harass, or harm. Your use of this generative AI service is subject to our Prohibited Use Policy (https://policies.google.com/terms/generative-ai/use-policy).

Provide an image and use text prompts to add, remove, or modify elements, change the style, or adjust the color grading.

The following example demonstrates uploading `base64` encoded images. For multiple images, larger payloads, and supported MIME types, check the Image understanding (/gemini-api/docs/image-understanding) page.

Python (#python) JavaScript (#javascript) Go (#go) Java (#java) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flas
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
        \"parts\":[
            {\"text\": \"'Create a picture of my cat eating a nano-banana
            {
              \"inline_data\": {
                \"mime_type\":\"image/jpeg\",
                \"data\": \"<BASE64_IMAGE_DATA>\"
              }
            }
        ]
      }]
    }"
```

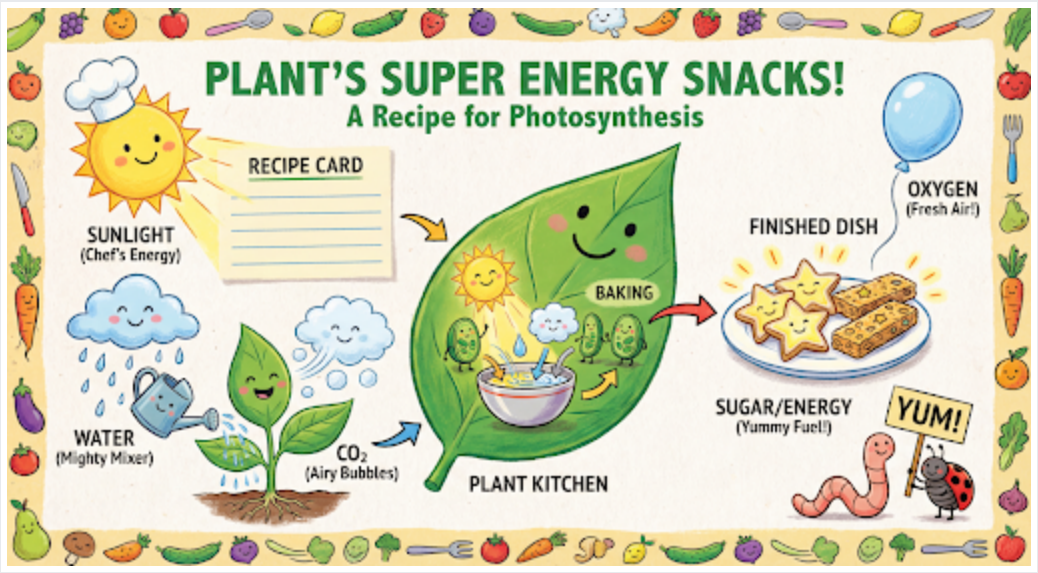
Multi-turn image editing

Keep generating and editing images conversationally. Chat or multi-turn conversation is the recommended way to iterate on images. The following example shows a prompt to generate an infographic about photosynthesis.

Python (#python) Javascript (#javascript) Go (#go) Java (#java) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
```

```
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "role": "user",
      "parts": [
        {"text": "Create a vibrant infographic that explains photosynthesi
      ]
    }],
    "generationConfig": {
      "responseModalities": ["TEXT", "IMAGE"]
    }
  }'
```



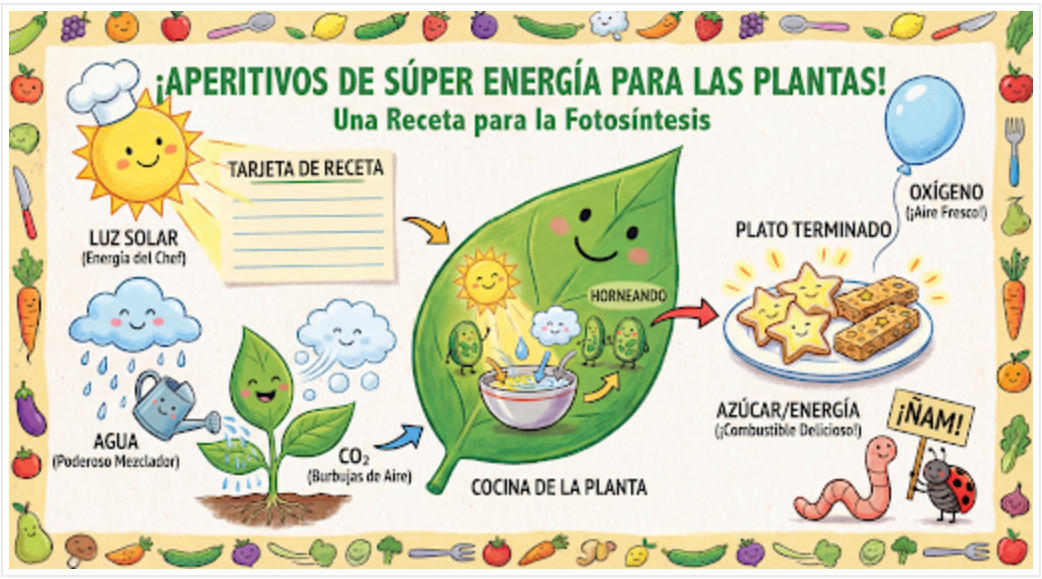
AI-generated infographic about photosynthesis

You can then use the same chat to change the language on the graphic to Spanish.

Python (#python) Javascript (#javascript) Go (#go) Java (#java) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H 'Content-Type: application/json' \
  -d '{
    "contents": [
      {
        "role": "user",
```

```
        "parts": [{"text": "Create a vibrant infographic that explains pho
      },
      {
        "role": "model",
        "parts": [{"inline_data": {"mime_type": "image/png", "data": "<PRE
      },
      {
        "role": "user",
        "parts": [{"text": "Update this infographic to be in Spanish. Do n
      }
    ],
    "tools": [{"google_search": {}}],
    "generationConfig": {
      "responseModalities": ["TEXT", "IMAGE"],
      "imageConfig": {
        "aspectRatio": "16:9",
        "imageSize": "2K"
      }
    }
  }'
```



AI-generated infographic of photosynthesis in Spanish

New with Gemini 3 Pro Image

Gemini 3 Pro Image (`gemini-3-pro-image-preview`) is a state-of-the-art image generation and editing model optimized for professional asset production. Designed to tackle the most

challenging workflows through advanced reasoning, it excels at complex, multi-turn creation and modification tasks.

- **High-resolution output**: Built-in generation capabilities for 1K, 2K, and 4K visuals.
- **Advanced text rendering**: Capable of generating legible, stylized text for infographics, menus, diagrams, and marketing assets.
- **Grounding with Google Search**: The model can use Google Search as a tool to verify facts and generate imagery based on real-time data (e.g., current weather maps, stock charts, recent events).
- **Thinking mode**: The model utilizes a "thinking" process to reason through complex prompts. It generates interim "thought images" (visible in the backend but not charged) to refine the composition before producing the final high-quality output.
- **Up to 14 reference images**: You can now mix up to 14 reference images to produce the final image.

Use up to 14 reference images

Gemini 3 Pro Preview lets you to mix up to 14 reference images. These 14 images can include the following:

- Up to 6 images of objects with high-fidelity to include in the final image
- Up to 5 images of humans to maintain character consistency

Python (#python) Javascript (#javascript) Go (#go) Java (#java) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
        \"parts\":[
            {\"text\": \"An office group photo of these people, they are ma
            {\"inline_data\": {\"mime_type\":\"image/png\", \"data\": \"<BA
            {\"inline_data\": {\"mime_type\":\"image/png\", \"data\": \"<BA
            {\"inline_data\": {\"mime_type\":\"image/png\", \"data\": \"<BA
```

```
            {\"inline_data\": {\"mime_type\":\"image/png\", \"data\": \"<B
            {\"inline_data\": {\"mime_type\":\"image/png\", \"data\": \"<B
        ]
      }],
      \"generationConfig\": {
        \"responseModalities\": [\"TEXT\", \"IMAGE\"],
        \"imageConfig\": {
          \"aspectRatio\": \"5:4\",
          \"imageSize\": \"2K\"
        }
      }
    }"
```

AI-generated office group photo

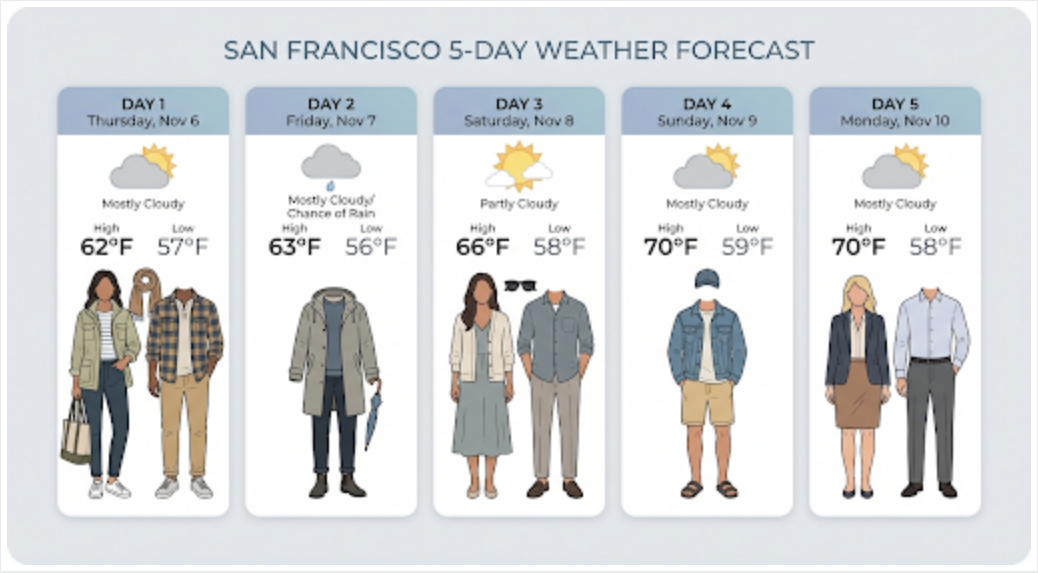
Grounding with Google Search

Use the Google Search tool (/gemini-api/docs/google-search) to generate images based on real-time information, such as weather forecasts, stock charts, or recent events.

Note that when using Grounding with Google Search with image generation, image-based search results are not passed to the generation model and are excluded from the response.

Python (#python)Javascript (#javascript)Java (#java)REST

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{"parts": [{"text": "Visualize the current weather forecas
    "tools": [{"google_search": {}}],
    "generationConfig": {
      "responseModalities": ["TEXT", "IMAGE"],
      "imageConfig": {"aspectRatio": "16:9"}
    }
  }'
```



AI-generated five day weather chart for San Francisco

The response includes **`groundingMetadata`** which contains the following required fields:

- **`searchEntryPoint`**: Contains the HTML and CSS to render the required search suggestions.
- **`groundingChunks`**: Returns the top 3 web sources used to ground the generated image

Generate images up to 4K resolution

Gemini 3 Pro Image generates 1K images by default but can also output 2K and 4K images. To generate higher resolution assets, specify the `image_size` in the `generation_config`.

You must use an uppercase 'K' (e.g., 1K, 2K, 4K). Lowercase parameters (e.g., 1k) will be rejected.

Python (#python) Javascript (#javascript) Go (#go) Java (#java) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{"parts": [{"text": "Da Vinci style anatomical sketch of a
    "tools": [{"google_search": {}}],
    "generationConfig": {
      "responseModalities": ["TEXT", "IMAGE"],
      "imageConfig": {"aspectRatio": "1:1", "imageSize": "1K"}
    }
  }'
```

The following is an example image generated from this prompt:

AI-generated Da Vinci style anatomical sketch of a dissected Monarch butterfly.

Thinking Process

The Gemini 3 Pro Image Preview model is a thinking model and uses a reasoning process ("Thinking") for complex prompts. This feature is enabled by default and cannot be disabled in the API. To learn more about the thinking process, see the Gemini Thinking (/gemini-api/docs/thinking) guide.

The model generates up to two interim images to test composition and logic. The last image within Thinking is also the final rendered image.

You can check the thoughts that lead to the final image being produced.

Python Javascript (#javascript) (#python)

```
for part in response.parts:
    if part.thought:
        if part.text:
            print(part.text)
        elif image:= part.as_image():
            image.show()
```

Thought Signatures

Thought signatures are encrypted representations of the model's internal thought process and are used to preserve reasoning context across multi-turn interactions. All responses include a `thought_signature` field. As a general rule, if you receive a thought signature in a model response, you should pass it back exactly as received when sending the conversation history in the next turn. Failure to circulate thought signatures may cause the response to fail. Check the thought signature (/gemini-api/docs/thought-signatures) documentation for more explanations of signatures overall.

Note: If you use the official Google Gen AI SDKs (/gemini-api/docs/libraries) and use the chat feature (or append the full model response object directly to history), **thought signatures are handled automatically**. You do not need to manually extract or manage them, or change your code.

Here is how thought signatures work:

- All `inline_data` parts with image `mimetype` which are part of the response should have signature.

- If there are some text parts at the beginning (before any image) right after the thoughts, the first text part should also have a signature.
- If `inline_data` parts with image `mimetype` are part of thoughts, they won't have signatures.

The following code shows an example of where thought signatures are included:

```
[
  {
    "inline_data": {
      "data": "<base64_image_data_0>",
      "mime_type": "image/png"
    },
    "thought": true // Thoughts don't have signatures
  },
  {
    "inline_data": {
      "data": "<base64_image_data_1>",
      "mime_type": "image/png"
    },
    "thought": true // Thoughts don't have signatures
  },
  {
    "inline_data": {
      "data": "<base64_image_data_2>",
      "mime_type": "image/png"
    },
    "thought": true // Thoughts don't have signatures
  },
  {
    "text": "Here is a step-by-step guide to baking macarons, presented in three
    "thought_signature": "<Signature_A>" // The first non-thought part always ha
  },
  {
    "inline_data": {
      "data": "<base64_image_data_3>",
      "mime_type": "image/png"
    },
    "thought_signature": "<Signature_B>" // All image parts have a signatures
  },
  {
    "text": "\n\n### Step 2: Baking and Developing Feet\n\nOnce piped, the macar
```

```
    // Follow-up text parts don't have signatures
  },
  {
    "inline_data": {
      "data": "<base64_image_data_4>",
      "mime_type": "image/png"
    },
    "thought_signature": "<Signature_C>" // All image parts have a signatures
  },
  {
    "text": "\n\n### Step 3: Assembling the Macaron\n\nThe final step is to pair
  },
  {
    "inline_data": {
      "data": "<base64_image_data_5>",
      "mime_type": "image/png"
    },
    "thought_signature": "<Signature_D>" // All image parts have a signatures
  }
]
```

Other image generation modes

Gemini supports other image interaction modes based on prompt structure and context, including:

- **Text to image(s) and text (interleaved):** Outputs images with related text.
  - Example prompt: "Generate an illustrated recipe for a paella."
- **Image(s) and text to image(s) and text (interleaved)**: Uses input images and text to create new related images and text.
  - Example prompt: (With an image of a furnished room) "What other color sofas would work in my space? can you update the image?"

Generate images in batch

If you need to generate a lot of images, you can use the Batch API (/gemini-api/docs/batch-api). You get higher rate limits (/gemini-api/docs/rate-limits) in exchange for a turnaround of up to 24 hours.

Check the Batch API image generation documentation (/gemini-api/docs/batch-api#image-generation) and the cookbook (https://colab.research.google.com/github/google-gemini/cookbook/blob/main/quickstarts/Batch_mode.ipynb) for Batch API image examples and code.

Prompting guide and strategies

Mastering image generation starts with one fundamental principle:

> **Describe the scene, don't just list keywords.** The model's core strength is its deep language understanding. A narrative, descriptive paragraph will almost always produce a better, more coherent image than a list of disconnected words.

Prompts for generating images

The following strategies will help you create effective prompts to generate exactly the images you're looking for.

1. Photorealistic scenes

For realistic images, use photography terms. Mention camera angles, lens types, lighting, and fine details to guide the model toward a photorealistic result.

Template (#template) Prompt (#prompt) Python (#python) Java (#java) JavaScript (#javascript) Go (#go) RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flasl
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
```

```
      "parts": [
        {"text": "A photorealistic close-up portrait of an elderly Japanes
      ]
    }]
  }'
```

A photorealistic close-up portrait of an elderly Japanese ceramicist...

2. Stylized illustrations & stickers

To create stickers, icons, or assets, be explicit about the style and request a transparent background.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flas
  -H "x-goog-api-key: $GEMINI_API_KEY" \
```

```
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "parts": [
        {"text": "A kawaii-style sticker of a happy red panda wearing a ti
      ]
    }]
  }'
```

A kawaii-style sticker of a happy red panda...

3. Accurate text in images

Gemini excels at rendering text. Be clear about the text, the font style (descriptively), and the overall design. Use Gemini 3 Pro Image Preview for professional asset production.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "parts": [
        {"text": "Create a modern, minimalist logo for a coffee shop calle
      ]
    }],
    "generationConfig": {
      "imageConfig": {
        "aspectRatio": "1:1"
      }
    }
  }'
```

Create a modern, minimalist logo for a coffee shop called 'The Daily Grind'...

4. Product mockups & commercial photography

Perfect for creating clean, professional product shots for ecommerce, advertising, or branding.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flasl
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "parts": [
        {"text": "A high-resolution, studio-lit product photograph of a mi
      ]
    }]
  }'
```

A high-resolution, studio-lit product photograph of a minimalist ceramic coffee mug...

5. Minimalist & negative space design

Excellent for creating backgrounds for websites, presentations, or marketing materials where text will be overlaid.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flasl
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "parts": [
        {"text": "A minimalist composition featuring a single, delicate re
      ]
    }]
  }'
```

A minimalist composition featuring a single, delicate red maple leaf...

6. Sequential art (Comic panel / Storyboard)

Builds on character consistency and scene description to create panels for visual storytelling. For accuracy with text and storytelling ability, these prompts work best with Gemini 3 Pro Image Preview.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "parts": [
        {"text": "Make a 3 panel comic in a gritty, noir art style with hig
        {"inline_data": {"mime_type": "image/jpeg", "data": "<BASE64_IMAGE_
      ]
    }]
  }'
```

Input	Output
	
	Make a 3 panel comic in a gritty, noir art style...

Input image

7. Grounding with Google Search

Use Google Search to generate images based on recent or real-time information. This is useful for news, weather, and other time-sensitive topics.

Prompt (#prompt) Python (#python) Java (#java) JavaScript (#javascript) Go (#go) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{"parts": [{"text": "Make a simple but stylish graphic of
    "tools": [{"google_search": {}}],
    "generationConfig": {
      "responseModalities": ["TEXT", "IMAGE"],
      "imageConfig": {"aspectRatio": "16:9"}
    }
  }'
```

AI-generated graphic of an Arsenal football score

Prompts for editing images

These examples show how to provide images alongside your text prompts for editing, composition, and style transfer.

1. Adding and removing elements

Provide an image and describe your change. The model will match the original image's style, lighting, and perspective.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flas
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
```

```
-d "{
  \"contents\": [{
    \"parts\":[
        {\"text\": \"Using the provided image of my cat, please add a s
        {
          \"inline_data\": {
            \"mime_type\":\"image/png\",
            \"data\": \"<BASE64_IMAGE_DATA>\"
          }
        }
    ]
  }]
}"
```

Input	Output
	
A photorealistic picture of a fluffy ginger cat...	*Using the provided image of my cat, please add a small, knitted wizard hat...*

2. Inpainting (Semantic masking)

Conversationally define a "mask" to edit a specific part of an image while leaving the rest untouched.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flas
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
        \"parts\":[
            {
              \"inline_data\": {
                \"mime_type\":\"image/png\",
                \"data\": \"<BASE64_IMAGE_DATA>\"
              }
            },
            {\"text\": \"Using the provided image of a living room, change
        ]
      }]
    }"
```

Input	Output
	
A wide shot of a modern, well-lit living room...	*Using the provided image of a living room, change only the blue sofa to be a vintage, brown leather chesterfield sofa...*

3. Style transfer

Provide an image and ask the model to recreate its content in a different artistic style.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flasl
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
        \"parts\":[
            {
              \"inline_data\": {
                \"mime_type\":\"image/png\",
                \"data\": \"<BASE64_IMAGE_DATA>\"
              }
            },
            {\"text\": \"Transform the provided photograph of a modern city
        ]
      }]
    }"
```

Input	Output
	
A photorealistic, high-resolution photograph of a busy city street...	*Transform the provided photograph of a modern city street at night...*

4. Advanced composition: Combining multiple images

Provide multiple images as context to create a new, composite scene. This is perfect for product mockups or creative collages.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flas
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
        \"parts\":[
            {
              \"inline_data\": {
                \"mime_type\":\"image/png\",
                \"data\": \"<BASE64_IMAGE_DATA_1>\"
              }
            },
            {
              \"inline_data\": {
                \"mime_type\":\"image/png\",
                \"data\": \"<BASE64_IMAGE_DATA_2>\"
              }
            },
            {\"text\": \"Create a professional e-commerce fashion photo. Ta
        ]
      }]
    }"
```

Input 1	Input 2	Output

A professionally shot photo of a blue floral summer dress...

Full-body shot of a woman with her hair in a bun...

Create a profession ph

5. High-fidelity detail preservation

To ensure critical details (like a face or logo) are preserved during an edit, describe them in great detail along with your edit request.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flas
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
        \"parts\":[
            {
              \"inline_data\": {
                \"mime_type\":\"image/png\",
                \"data\": \"<BASE64_IMAGE_DATA_1>\"
              }
            },
            {
              \"inline_data\": {
                \"mime_type\":\"image/png\",
                \"data\": \"<BASE64_IMAGE_DATA_2>\"
```

```
              }
            },
            {\"text\": \"Take the first image of the woman with brown hair
        ]
      }]
    }"
```

Input 2	Output
	
nal headshot of a woman with wn hair and blue eyes...	*A simple, modern logo with the letters 'G' and 'A'...*
	
Take the first image of the wom brown hair, blue eyes, and a n expression...	

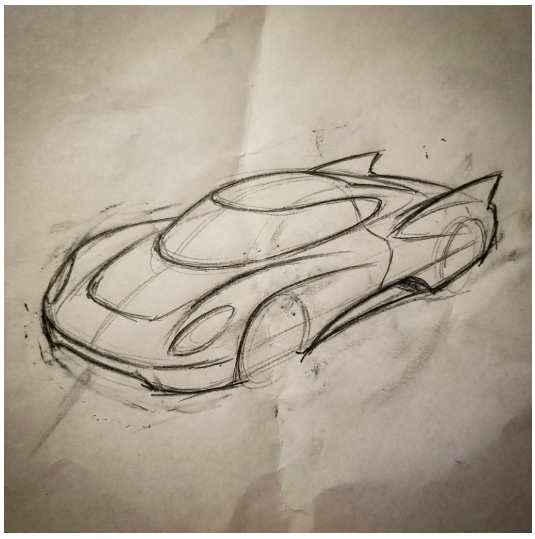
6. Bring something to life

Upload a rough sketch or drawing and ask the model to refine it into a finished image.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
```

```
          \"parts\":[
              {
                \"inline_data\": {
                  \"mime_type\":\"image/png\",
                  \"data\": \"<BASE64_IMAGE_DATA>\"
                }
              },
              {\"text\": \"Turn this rough pencil sketch of a futuristic car
          ]
        }]
      }"
```

Input	Output
	
Rough sketch of a car	*Polished photo of a car*

7. Character consistency: 360 view

You can generate 360-degree views of a character by iteratively prompting for different angles. For best results, include previously generated images in subsequent prompts to maintain consistency. For complex poses, include a reference image of the desired pose.

Template (#template)Prompt (#prompt)Python (#python)Java (#java)JavaScript (#javascript)Go (#go)RES

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
    -H "x-goog-api-key: $GEMINI_API_KEY" \
    -H 'Content-Type: application/json' \
    -d "{
      \"contents\": [{
        \"parts\":[
            {\"text\": \"A studio portrait of this man against white, in p
            {
              \"inline_data\": {
                \"mime_type\":\"image/jpeg\",
                \"data\": \"<BASE64_IMAGE_DATA>\"
              }
            }
        ]
      }]
    }"
```

Input	Output 1	Output 2
		
Original image	*Man in white glasses looking right*	*Man in white glas*

Best Practices

To elevate your results from good to great, incorporate these professional strategies into your workflow.

- **Be Hyper-Specific:** The more detail you provide, the more control you have. Instead of "fantasy armor," describe it: "ornate elven plate armor, etched with silver leaf patterns, with a high collar and pauldrons shaped like falcon wings."
- **Provide Context and Intent:** Explain the *purpose* of the image. The model's understanding of context will influence the final output. For example, "Create a logo for a high-end, minimalist skincare brand" will yield better results than just "Create a logo."
- **Iterate and Refine:** Don't expect a perfect image on the first try. Use the conversational nature of the model to make small changes. Follow up with prompts like, "That's great, but can you make the lighting a bit warmer?" or "Keep everything the same, but change the character's expression to be more serious."
- **Use Step-by-Step Instructions:** For complex scenes with many elements, break your prompt into steps. "First, create a background of a serene, misty forest at dawn. Then, in the foreground, add a moss-covered ancient stone altar. Finally, place a single, glowing sword on top of the altar."
- **Use "Semantic Negative Prompts":** Instead of saying "no cars," describe the desired scene positively: "an empty, deserted street with no signs of traffic."
- **Control the Camera:** Use photographic and cinematic language to control the composition. Terms like `wide-angle shot`, `macro shot`, `low-angle perspective`.

Limitations

- For best performance, use the following languages: EN, ar-EG, de-DE, es-MX, fr-FR, hi-IN, id-ID, it-IT, ja-JP, ko-KR, pt-BR, ru-RU, ua-UA, vi-VN, zh-CN.
- Image generation does not support audio or video inputs.
- The model won't always follow the exact number of image outputs that the user explicitly asks for.
- `gemini-2.5-flash-image` works best with up to 3 images as input, while `gemini-3-pro-image-preview` supports 5 images with high fidelity, and up to 14 images in total.

- When generating text for an image, Gemini works best if you first generate the text and then ask for an image with the text.
- All generated images include a SynthID watermark (/responsible/docs/safeguards/synthid).

Optional configurations

You can optionally configure the response modalities and aspect ratio of the model's output in the `config` field of `generate_content` calls.

Output types

The model defaults to returning text and image responses (i.e. `response_modalities=['Text', 'Image']`). You can configure the response to return only images without text using `response_modalities=['Image']`.

Python (#python) JavaScript (#javascript) Go (#go) Java (#java) REST (#rest)

```
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flasl
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "contents": [{
      "parts": [
        {"text": "Create a picture of a nano banana dish in a fancy restau
      ]
    }],
    "generationConfig": {
      "responseModalities": ["Image"]
    }
  }'
```

Aspect ratios and image size

The model defaults to matching the output image size to that of your input image, or otherwise generates 1:1 squares. You can control the aspect ratio of the output image using the `aspect_ratio` field under `image_config` in the response request, shown here:

Python (#python) JavaScript (#javascript) Go (#go) Java (#java) REST (#rest)

```
# For gemini-2.5-flash-image
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H 'Content-Type: application/json' \
  -d '{
    "contents": [{
      "parts": [
        {"text": "Create a picture of a nano banana dish in a fancy restau
      ]
    }],
    "generationConfig": {
      "imageConfig": {
        "aspectRatio": "16:9"
      }
    }
  }'

# For gemini-3-pro-image-preview
curl -s -X POST \
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-3-pro-ima
  -H "x-goog-api-key: $GEMINI_API_KEY" \
  -H 'Content-Type: application/json' \
  -d '{
    "contents": [{
      "parts": [
        {"text": "Create a picture of a nano banana dish in a fancy restau
      ]
    }],
    "generationConfig": {
      "imageConfig": {
        "aspectRatio": "16:9",
        "imageSize": "2K"
      }
    }
```

```
    }'
```

The different ratios available and the size of the image generated are listed in the following tables:

Gemini 2.5 Flash Image

Aspect ratio	Resolution	Tokens
1:1	1024x1024	1290
2:3	832x1248	1290
3:2	1248x832	1290
3:4	864x1184	1290
4:3	1184x864	1290
4:5	896x1152	1290
5:4	1152x896	1290
9:16	768x1344	1290
16:9	1344x768	1290
21:9	1536x672	1290

Gemini 3 Pro Image Preview

Aspect ratio	1K resolution	1K tokens	2K resolution	2K tokens	4K resolution	4K tokens
1:1	1024x1024	1120	2048x2048	1120	4096x4096	2000
2:3	848x1264	1120	1696x2528	1120	3392x5056	2000
3:2	1264x848	1120	2528x1696	1120	5056x3392	2000
3:4	896x1200	1120	1792x2400	1120	3584x4800	2000
4:3	1200x896	1120	2400x1792	1120	4800x3584	2000
4:5	928x1152	1120	1856x2304	1120	3712x4608	2000
5:4	1152x928	1120	2304x1856	1120	4608x3712	2000
9:16	768x1376	1120	1536x2752	1120	3072x5504	2000
16:9	1376x768	1120	2752x1536	1120	5504x3072	2000
21:9	1584x672	1120	3168x1344	1120	6336x2688	2000

Model selection

Choose the model best suited for your specific use case.

- **Gemini 3 Pro Image Preview (Nano Banana Pro Preview)** is designed for professional asset production and complex instructions. This model features real-world grounding using Google Search, a default "Thinking" process that refines composition prior to generation, and can generate images of up to 4K resolutions. Check the model pricing and capabilities (/gemini-api/docs/models/gemini#gemini-3-pro-image-preview) page for more details.

- **Gemini 2.5 Flash Image (Nano Banana)** is designed for speed and efficiency. This model is optimized for high-volume, low-latency tasks and generates images at 1024px resolution. Check the model pricing and capabilities (/gemini-api/docs/models/gemini#gemini-2.5-flash-image) page for more details.

When to use Imagen

In addition to using Gemini's built-in image generation capabilities, you can also access Imagen (/gemini-api/docs/imagen), our specialized image generation model, through the Gemini API.

Imagen 4 should be your go-to model when starting to generate images with Imagen. Choose Imagen 4 Ultra for advanced use-cases or when you need the best image quality (note that can only generate one image at a time).

What's next

- Find more examples and code samples in the cookbook guide (https://colab.research.google.com/github/google-gemini/cookbook/blob/main/quickstarts/Get_Started_Nano_Banana.ipynb).
- Check out the Veo guide (/gemini-api/docs/video) to learn how to generate videos with the Gemini API.
- To learn more about Gemini models, see Gemini models (/gemini-api/docs/models/gemini).

Last updated 2025-12-28 UTC.