

```
a=10
b=20
c=a+b
c
```

30

✓ Step:1 Install the required libraries

```
!pip install google-generativeai
```

```
Requirement already satisfied: google-generativeai in /usr/local/lib/python3.10/dist-packages (0.3.2)
Requirement already satisfied: google-ai-generativelanguage==0.4.0 in /usr/local/lib/python3.10/dist-packages (from google-generativeai)
Requirement already satisfied: google-auth in /usr/local/lib/python3.10/dist-packages (from google-generativeai) (2.27.0)
Requirement already satisfied: google-api-core in /usr/local/lib/python3.10/dist-packages (from google-generativeai) (2.11.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from google-generativeai) (4.10.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packages (from google-generativeai) (3.20.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from google-generativeai) (4.66.2)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in /usr/local/lib/python3.10/dist-packages (from google-ai-generativelanguage)
Requirement already satisfied: googleapis-common-protos<2.0.dev0,>=1.56.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai)
Requirement already satisfied: requests<3.0.0.dev0,>=2.18.0 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth->google-generativeai)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth->google-generativeai)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth->google-generativeai) (4.9)
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai)
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-auth)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,>=2.18.0->google-generativeai)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,>=2.18.0->google-api-core)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,>=2.18.0->google-generativeai)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0.dev0,>=2.18.0->google-generativeai)
```

✓ Step:2 API configuration

```
import google.generativeai as genai
api_key = "AIzaSyCS3hjhb_Mvl3gvzf44XSGMfZwZCrHfAN0"
genai.configure(api_key=api_key)
```

✓ Step:3 Check out the no of models

```
for m in genai.list_models():
    print(m.name)

models/chat-bison-001
models/text-bison-001
models/embedding-gecko-001
models/gemini-1.0-pro
models/gemini-1.0-pro-001
models/gemini-1.0-pro-latest
models/gemini-1.0-pro-vision-latest
models/gemini-pro
models/gemini-pro-vision
models/embedding-001
models/aqa
```

Text model ---> Gemini pro



Image model ---> Gemini-pro-vision

✓ Step:4 Gemini pro

```
model = genai.GenerativeModel("gemini-pro")
model
```

```
genai.GenerativeModel(
  model_name='models/gemini-pro',
  generation_config={},
  safety_settings={}
)
```

```
response = model.generate_content("Who is PM of India")
print(response.text)
```

Narendra Modi

Double-click (or enter) to edit

✓ Step:5 Image model

```
#Gemini-pro-vision
```

```
model = genai.GenerativeModel("gemini-pro-vision")
model
```

```
genai.GenerativeModel(
  model_name='models/gemini-pro-vision',
  generation_config={},
  safety_settings={}
)
```

```
from PIL import Image #Pillow
img = Image.open("/content/IMG-20240318-WA0023.jpg")
img
```



```
model = genai.GenerativeModel("gemini-pro-vision")
prompt = "What are the objects given in above image"
response = model.generate_content([prompt,img])
print(response.text)
```

Two white butterflies on a stalk with yellow flowers.



```
model = genai.GenerativeModel("gemini-pro-vision")
prompt = "Can you provide the caption on the image"
response = model.generate_content([prompt,img])
print(response.parts)
```

```
[text: " Two white butterflies on a stalk with yellow flowers."
]
```

✓ Create a chat

```
model = genai.GenerativeModel("gemini-pro")
chat = model.start_chat(history=[])
question = "Who is PM of India"
response = chat.send_message(question)
response.text
```

'Narendra Modi'

```
chat.history
```

```
[parts {
  text: "Who is PM of India"
}
role: "user",
parts {
  text: "Narendra Modi"
}
role: "model"]
```

```
question = "He belongs to which party?"
response = chat.send_message(question)
response.text
```

'Bharatiya Janata Party (BJP)'

```
question = "When it's established?"
response = chat.send_message(question)
response.text
```

'April 6, 1980'

```
chat.history
```

```
[parts {
  text: "Who is PM of India"
}
role: "user",
parts {
  text: "Narendra Modi"
}
role: "model",
parts {
  text: "He belongs to which party?"
}
role: "user",
parts {
  text: "Bharatiya Janata Party (BJP)"
}
role: "model",
parts {
  text: "When it's established?"
}
role: "user",
parts {
  text: "April 6, 1980"
}
role: "model"]
```

```
#Create a simple chatbot application
model = genai.GenerativeModel("gemini-pro")
chat = model.start_chat(history=[])
while True:
    question = input("YOU: Ask a question")
    if question=="exit":
        break
    response = chat.send_message(question)
    print("Bot:",response.text)
```



YOU: Ask a questionHow is the day
 Bot: As an AI language model, I don't have personal experiences or the ability to perceive
 YOU: Ask a questionHello
 Bot: Hello there! How may I assist you today?
 YOU: Ask a questionwrite a poem on kid
 Bot: **Ode to a Child**

A bundle of joy, a precious soul,
 A tiny human, yet a spirit whole.
 With eyes that sparkle, like stars so bright,
 And laughter that fills the heart with light.

Their tiny hands, so soft and small,
 Reach out to the world, eager to crawl.
 Their feet, like petals, dance with glee,
 As they explore their surroundings, so free.

Their minds, like sponges, absorb it all,
 From the colors of rainbows to the songs of the thrall.
 They learn and grow with each passing day,
 Their curiosity knows no bounds, they say.

Their hearts, so pure and full of love,
 Embrace the world with open arms, like a dove.
 They trust and believe with unwavering faith,
 That everything is possible, and dreams can't be erased.

Oh, precious child, our future's hope,
 May your life be filled with joy and scope.
 May you soar high, on wings of your own,
 And make the world a better place, like a seed that's sown.

So cherish each moment, as time flies fast,
 And watch in wonder, as your child surpasses.
 For in their eyes, the future lies,
 A world of possibilities, where their dreams will rise.
 YOU: Ask a questionexit

```
!pip install streamlit
!npm install localtunnel
```

```
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.10/dist-packages (from streamlit) (2.31.0)
Requirement already satisfied: rich<14,>=10.14.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (13.7.1)
Requirement already satisfied: tenacity<9,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (8.2.3)
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.10.0)
Collecting gitpython!=3.1.19,<4,>=3.0.7 (from streamlit)
  Downloading GitPython-3.1.43-py3-none-any.whl (207 kB)
    207.3/207.3 kB 23.5 MB/s eta 0:00:00
Collecting pydeck<1,>=0.8.0b4 (from streamlit)
  Downloading pydeck-0.8.1b0-py2.py3-none-any.whl (4.8 MB)
    4.8/4.8 MB 29.1 MB/s eta 0:00:00
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python3.10/dist-packages (from streamlit) (6.3.3)
Collecting watchdog<2.1.5 (from streamlit)
  Downloading watchdog-4.0.0-py3-none-manylinux2014_x86_64.whl (82 kB)
    83.0/83.0 kB 10.8 MB/s eta 0:00:00
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (0.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (3.1.3)
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (4.19.2)
Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (0.12.1)
Collecting gitdb<5,>=4.0.1 (from gitpython!=3.1.19,<4,>=3.0.7->streamlit)
  Downloading gitdb-4.0.11-py3-none-any.whl (62 kB)
    62.7/62.7 kB 7.8 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2024.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (3.3.2)
```

```
recompiling tensorflow packaging
```

```
Found existing installation: packaging 24.0
```

```
Uninstalling packaging-24.0:
```

```
Successfully uninstalled packaging-24.0
```

```
Successfully installed gitdb-4.0.11 gitpython-3.1.43 packaging-23.2 pydeck-0.8.1b0 smmap-5.0.1 streamlit-1.32.2 watchdog-4.0.0
```

```
npm ERR! code E404
```

```
npm ERR! 404 Not Found - GET https://registry.npmjs.org/localtunnel - Not found
```

```
npm ERR! 404
```

```
npm ERR! 404 'localtunnel@latest' is not in the npm registry.
```

```
npm ERR! 404 You should bug the author to publish it (or use the name yourself!)
```

```
npm ERR! 404
```

```
npm ERR! 404 Note that you can also install from a
```

```
npm ERR! 404 tarball, folder, http url, or git url.
```

```
npm ERR! A complete log of this run can be found in:
```

```
npm ERR! /root/.npm/_logs/2024-04-04T06_03_33_350Z-debug.log
```

```
%%writefile app.py
import streamlit as st
import google.generativeai as genai

api_key = "AIzaSyCS3hjhb_Mvl3gvzf44XSGMfZWZCrHfAN0"
genai.configure(api_key= api_key)

#choose the heading
st.header("Gemini AI Chat Bot")

print("Input")

question = st.text_input("Input ")

print("Click for response")

### Use genai skill ###

if st.button("RESPONSE"):
    model = genai.GenerativeModel("gemini-pro")
    chat = model.start_chat(history=[])
    response = chat.send_message(question)
    print(response)
    st.markdown(response.text)
```

Overwriting app.py

```
!streamlit run /content/app.py & npx localtunnel --port 8501 & curl ipv4.icanhazip.com
```

➡ 34.106.202.172

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: <http://172.28.0.12:8501>

External URL: <http://34.106.202.172:8501>

npx: installed 22 in 2.392s

your url is: <https://violet-rockets-sniff.loca.lt>

Input

Click for response

Input

Click for response

Input

Click for response

<google.generativeai.types.generation_types.GenerateContentResponse object at 0x7fa3290d8a60>

Stopping...

