

MORE CHAT TOOLS: WEB SEARCH, CODE EXECUTION

YALE SOM

PROFESSOR TAUHID ZAMAN



Overview

- **Web search tools**
- **Code execution tools**
- **Downloading tweets (new Chrome plugin)**

Web Search Tools

- **What they do**
 - **Grounding: connects the AI to Google Search**
 - **Freshness: accesses news, social trends, and real-time data**
 - **Verification: provides citations/links to sources**

How To Use (API)

- Tool definition:

```
const SEARCH_TOOL = { googleSearch: {} };
```

- Tool call in API:

```
export const streamChat = async function* (history, newMessage, imageParts = [], useCodeExecution = false) {  
  const systemInstruction = await loadSystemPrompt();  
  const tools = useCodeExecution ? [CODE_EXEC_TOOL] : [SEARCH_TOOL];  
  const model = genAI.getGenerativeModel({  
    model: MODEL,  
    tools,  
  });  
};
```

- Tool mention in system prompt

```
You have access to Google Search.  
Use it to verify facts,  
find current events, or provide up-to-date  
technical documentation.
```

Cost

- **Google searches with Gemini cost ~ \$0.002 per search**

Gemini API Pricing Guide (2026)

Model	Input (per 1M tokens)	Output (per 1M tokens)	Search Grounding Cost*
Gemini 3 Flash	\$0.50	\$3.00	\$14 / 1k queries
Gemini 3 Pro	\$2.00	\$12.00	\$25 / 1k queries
Gemini 2.5 Flash	\$0.30	\$2.50	\$35 / 1k queries
Gemini 1.5 Pro	\$1.25	\$5.00	\$35 / 1k queries

Code Execution Tools

- **What it does**
 - **Writes Python code**
 - **Runs code in a secure sandbox**
 - **Use code output to answer your question**

Code Execution Tool Steps

- **Prompt to Gemini:** *I have a CSV with columns timestamp and likes. Can you calculate the correlation between the hour of the day and the number of likes, then tell me the best time to post?*
- **Model's Generated Code (Executable Code)**

```
Python

import pandas as pd
import numpy as np

# 1. Load the data
df = pd.read_csv('engagement_data.csv')

# 2. Extract hour from timestamp
df['hour'] = pd.to_datetime(df['timestamp']).dt.hour

# 3. Calculate Correlation
correlation = df['hour'].corr(df['likes'])

# 4. Find peak performance hour
peak_hour = df.groupby('hour')['likes'].mean().idxmax()

print(f"Correlation: {correlation:.4f}")
print(f"Peak Hour: {peak_hour}")
```

- **Output from sandbox:**

```
Correlation: 0.6421
Peak Hour: 19
```

- **Gemini output:** *Based on the code execution, there is a moderate positive correlation (0.64) between the hour of the day and likes. My analysis shows that your posts perform best at 7:00 PM (19:00). I recommend scheduling your high-value content for that window.*

How to Use (API)

- Tool definition:

```
const CODE_EXEC_TOOL = { codeExecution: {} };  
  
export const CODE_KEYWORDS = /\b(plot|chart|graph|analyz|statistic|regression|correlat
```

- Tool call in API:

```
export const streamChat = async function* (history, newMessage, imageParts = [], useCodeExecution = false) {  
  const systemInstruction = await loadSystemPrompt();  
  const tools = useCodeExecution ? [CODE_EXEC_TOOL] : [SEARCH_TOOL];  
  const model = genAI.getGenerativeModel({  
    model: MODEL,  
    tools,  
  });  
};
```

- Tool mention in system prompt

```
Use code execution (Python) only when the student asks for a  
plot, chart, or something the tools above can't handle.
```


The Power of Code Execution: Iterative Intelligence

- **The "Think-Execute-Fix" Loop:** Unlike a standard chat, the Code Execution tool doesn't just "guess" an answer. It follows a multi-step workflow:
 - Reasoning: Gemini identifies that the request requires math or data manipulation.
 - Generation: It writes a Python script using libraries like Pandas, NumPy, or Matplotlib.
 - Self-Correction: If the code fails (e.g., a missing column or a typo), Gemini sees the error message and automatically rewrites the code. It can try up to 5 times before giving up.
- **Built-in Data Science Toolkit:** the sandbox comes pre-loaded with the industry-standard libraries:
 - Pandas: For cleaning messy social media exports.
 - Matplotlib: For generating charts (scatter plots, histograms) on the fly.
 - NumPy: For high-performance statistical calculations.



Downloading Tweets

- TwExtract - new Chrome plugin that works as of February 18th, 2026

The image displays the Chrome Web Store page for the TwExtract extension. The page title is "TwExtract - export anyone's tweets to CSV". It shows a rating of 3.0 stars from 14 reviews and is categorized as a "Tools" extension with 2,000 users. Below the main title, there is a diagram illustrating the process: a Twitter logo icon is followed by three blue arrows pointing right, which then point to a green document icon labeled "CSV". Below this diagram, the text reads "Extract all tweets from a user to CSV/Excel". To the right of the diagram, there is a screenshot of a Twitter profile page for "DSG Toast (@DisguisedToast)". A red arrow points to the username "@DisguisedToast" in the bio, with a red text overlay that says "1. Copy the Twitter username".

Coding Session

- **Clone an improved chat app**
- **Push to our own repo**
- **Host on Render**
- **Test the chatbot on a csv file of tweets**
- **Add some tools and features if we have time**