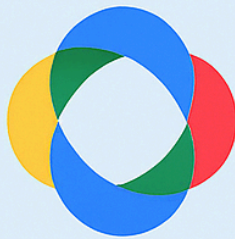


Build AI-Powered Google Sheets with Gemini: A Complete Beginner's Guide

Build AI-Powered Google Sheets with Gemini



- Custom function
- Sheet automations
- Setup-function
- Step-by-step guide

=	=GEMINI("prompt")	
A	A	
1	Ideas	
2	Develop an app	
3	Write a blog post	
4	Create a podcast	
5	Design a website	
6		

 Build AI-Powered Google Sheets with Gemini: A Complete Beginner's Guide	1
Learn how to use Google Apps Script + Gemini to automate ideas, translations, emails, summaries, and more	2
 What This Project Does	3
✓ A custom function	3
✓ Eight ready-to-use AI automations	3
✓ A setup function to auto-create sample sheets	3

 How the Code Is Structured	3
① Shared Helpers (Behind the Scenes)	3
getGeminiApiKey_()	3
callGemini_(prompt)	3
② Custom Function: =GEMINI()	4
③ Sheet Automations (The Fun Part)	4
🧠 Example 1 — Generate Ideas	4
✍️ Example 2 — One Idea Per Row	4
📊 Example 3 — Explain a Formula	4
🌐 Example 4 — Translate a Column	4
📋 Example 5 — Generate TODOs from Notes	5
✉️ Example 6 — AI Email Writer	5
📝 Example 7 — Meeting Summary	5
👩🏫 Example 8 — Tutor Answer	5
 Setup Function (Beginner-Friendly Magic)	5
1. Ideas_Emails	5
2. Translate_Summary	5
3. Tutor	6
 How to Use the Script (Step-by-Step)	6
1. Open a Google Sheet	6
2. Paste the provided Code.gs into your project	6
3. Add your Gemini API key	6
4. Run	6
5. Start running the examples from the menu or editor	6
 Who This Is Perfect For	6
 Final Thoughts	6
① Setup steps (very quick)	7
② Full Code.gs for Google Sheets + Gemini	8
🔧 Script to populate data for all exercises	17
🔄 How to use this with your existing functions	21

Learn how to use Google Apps Script + Gemini to automate ideas, translations, emails, summaries, and more

AI is no longer something we only use in chat apps—it can now work directly **inside your Google Sheets**. With just a few lines of Apps Script, you can get Google's **Gemini** model to:

- ✨ Generate creative ideas
- ✨ Write emails for you
- ✨ Translate text
- ✨ Summarize meeting notes
- ✨ Explain formulas
- ✨ Tutor you on technical questions
- ✨ And even populate Sheets with auto-generated data

In this beginner-friendly guide, you'll learn exactly how the provided Code .gs file works, why each piece matters, and how to use it even if you've **never written code before**.

What This Project Does

This single script transforms Google Sheets into an AI-powered workspace.
It gives you:

✓ **A custom function**

Works like any other Sheets formula:

```
=GEMINI("Explain JavaScript closures in simple words")
```

✓ **Eight ready-to-use AI automations**

For example:

- Generate ideas based on a topic in A1
- Produce one idea per row
- Create TODO items from meeting notes
- Translate an entire column
- Summarize meetings
- Generate email drafts
- Produce step-by-step explanations
- Act as a tutor

✓ **A setup function to auto-create sample sheets**

Run once, and it builds 3 Sheets with example data so you can try everything immediately.

How the Code Is Structured

The script is designed for complete beginners, so everything is in **one single file**, divided into clear sections:

1 Shared Helpers (Behind the Scenes)

These functions are the “engine” that talk to the Gemini API.

getGeminiApiKey_()

Reads your API key from the Script properties.

This keeps your key **safe and hidden** in the project.

callGemini_(prompt)

Sends your prompt to Gemini and returns the AI's text response.

This function handles:

- Building the correct API URL

- Packaging your prompt into JSON
- Sending it over the internet
- Reading Gemini's answer
- Throwing helpful errors if something goes wrong

You never need to modify this—it's reusable across all your projects.

2 Custom Function: =GEMINI ()

This lets you use Gemini **like any other spreadsheet formula**.

Example:

```
=GEMINI("Write 3 funny facts about the moon")
```

Whatever Gemini replies will appear directly in the cell.

This is the easiest way for beginners to start using AI in Sheets.

3 Sheet Automations (The Fun Part)

These functions operate on your Sheets and use Gemini behind the scenes.

Example 1 — Generate Ideas

Topic goes in **A1**.

Gemini's ideas appear in **A2**.

```
function sheets_topicIdeas() {
  const sheet = getActiveSheetOrError_();
  const topic = sheet.getRange('A1').getValue();
  const ideas = callGemini_("Give me 5 creative ideas about: " +
topic);
  sheet.getRange('A2').setValue(ideas);
}
```

Perfect for brainstorming.

Example 2 — One Idea Per Row

Takes the same topic but formats each idea in a separate row.

This is useful for lists, planning, or sorting later.

Example 3 — Explain a Formula

Click on any cell containing a formula → run this function →

Gemini explains what the formula does **in plain English**.

Perfect for learning Sheets or debugging confusing spreadsheets.

Example 4 — Translate a Column

- Takes all text in A2 : A

- Asks you what language you want
- Writes the translated text into B2 :B

Great for international teams, ESL students, or language practice.

Example 5 — Generate TODOs from Notes

Give it meeting notes → get back a clean checklist.

This helps you convert rough notes into actionable tasks.

Example 6 — AI Email Writer

Put into A1/B1/C1:

- Topic
- Recipient name
- Extra context

Gemini writes a polished email for you in **D1**.

Example 7 — Meeting Summary

Takes notes from A2:A10 → writes a summary paragraph + bullets into B1.

Ideal for staff meetings, class notes, or team updates.

Example 8 — Tutor Answer

Put any question in A1 and Gemini gives you a **step-by-step explanation**.

Excellent for learning programming, math, or technical subjects.

Setup Function (Beginner-Friendly Magic)

Running:

```
setupGeminiExampleData();
```

Creates three sheets for you:

1. Ideas_Emails

Holds:

- Topic
- Recipient
- Context
- Sample meeting notes

2. Translate_Summary

Holds:

- Sentences to translate
- Meeting notes for summaries

3. Tutor

Holds:

- A question for the AI tutor

This means you don't have to type anything—every example “just works.”



How to Use the Script (Step-by-Step)

1. Open a Google Sheet

Click: **Extensions** → **Apps Script**

2. Paste the provided Code.gs into your project

3. Add your Gemini API key

In Apps Script:

Project Settings → **Script properties** → **Add property**

- Key: GEMINI_API_KEY
- Value: your API key

4. Run

```
setupGeminiExampleData()
```

Authorize when prompted.

5. Start running the examples from the menu or editor

That's it!

You now have a fully AI-enabled spreadsheet.



Who This Is Perfect For

This project is ideal for:

- Students learning JavaScript or Google Sheets
- Professionals wanting to automate tedious work
- Teachers looking to integrate AI into their workflow
- Developers wanting reusable Gemini helper functions
- Anyone who wants AI *inside* their spreadsheets

You don't need to know coding—each example is isolated and easy to run.



Final Thoughts

This script turns Google Sheets into a **powerful AI workspace**.

Once you learn the structure:

- Helpers → talk to Gemini
- Functions → do the useful actions
- Sheets → show the results

...you can build anything you imagine.

Below is the **full Code.gs** for Google Sheets:

- Includes the Gemini helpers
- Stores/reads your API key from Script Properties
- Provides:
 - =GEMINI () custom function
 - Topic → ideas
 - Ideas per row
 - Explain formula
 - Translate column
 - Extract TODOs
 - Email draft
 - Meeting summary
 - Tutor answer
- Includes a simple test runner

1 Setup steps (very quick)

1. Open a **Google Sheet**
→ **Extensions** → **Apps Script**
2. Delete any sample code and paste EVERYTHING below into Code . gs.
3. Add your Gemini API key:
 - In the Script Editor, go to **Project Settings** → **Script properties** → **Add script property**
 - Key: GEMINI_API_KEY

- Value: your key
 - Save.
4. Back in the editor, run something simple like `test_GEMINI()` once and accept permissions.

2 Full Code .gs for Google Sheets + Gemini

```
/**
 * Gemini + Google Sheets Examples
 *
 * SETUP:
 * 1. Bind this script to a Google Sheet (Extensions → Apps Script).
 * 2. In the Script Editor, go to Project Settings → Script
properties.
 * 3. Add:
 *     KEY:   GEMINI_API_KEY
 *     VALUE: your Gemini API key
 * 4. Save and then run one of the test functions (e.g., test_GEMINI).
 */

const GEMINI_MODEL_ID = 'gemini-2.5-flash';

/**
=====
 * SHARED HELPERS
 *
=====
 */

/**
 * Get Gemini API key from Script Properties.
 */
function getGeminiApiKey_() {
  const scriptProps = PropertiesService.getScriptProperties();
  const key = scriptProps.getProperty('GEMINI_API_KEY');
  if (!key) {
    throw new Error(
      'GEMINI_API_KEY is not set.\n' +
      'In the Script Editor, open Project Settings → Script
properties and add it.'
    );
  }
  return key;
}
```



```

}

/**
 * Core helper: send a text prompt to Gemini and return a text
response.
 *
 * @param {string} prompt
 * @returns {string} Plain text response from Gemini.
 */
function callGemini_(prompt) {
  const apiKey = getGeminiApiKey_();

  const url =
    'https://generativelanguage.googleapis.com/v1beta/models/' +
    GEMINI_MODEL_ID +
    ':generateContent?key=' +
    apiKey;

  const payload = {
    contents: [{ parts: [{ text: prompt }] }]
  };

  const res = UrlFetchApp.fetch(url, {
    method: 'post',
    contentType: 'application/json',
    payload: JSON.stringify(payload),
    muteHttpExceptions: true
  });

  const text = res.getContentText();
  const json = JSON.parse(text);

  try {
    return json.candidates[0].content.parts[0].text;
  } catch (e) {
    throw new Error('Unexpected Gemini response: ' + text);
  }
}

/**
 * Get active sheet or throw a clear error if this isn't bound to a
Sheet.
 */
function getActiveSheetOrError_() {

```

```

const ss = SpreadsheetApp.getActiveSpreadsheet();
if (!ss) {
  throw new Error(
    'No active spreadsheet found.\n' +
    'Make sure this script is bound to a Google Sheet (Extensions
→ Apps Script).');
};
}
return ss.getActiveSheet() || ss.getSheets()[0];
}

/*
=====
* CUSTOM FUNCTION
*
=====
*/

/**
 * =GEMINI("Write a haiku about coding.")
 *
 * @param {string} prompt
 * @return {string}
 * @customfunction
 */
function GEMINI(prompt) {
  if (typeof prompt !== 'string') {
    return 'Prompt must be a string.';
  }
  return callGemini_(prompt);
}

/**
 * Simple test for GEMINI() from the editor.
 */
function test_GEMINI() {
  const out = GEMINI('Write a short motivational message for new
coders.');
```

Logger.log(out);

```

}

/*
=====
* EXAMPLE 1 - A1 topic → A2 ideas (single cell)

```

```

*
=====
*/

function sheets_topicIdeas() {
  const sheet = getActiveSheetOrError_();
  const topic = sheet.getRange('A1').getValue();

  if (!topic) {
    sheet.getRange('A2').setValue('Please enter a topic in A1.');
```

return;

```
  }

  const ideas = callGemini_('Give me 5 creative ideas about: ' +
topic);
  sheet.getRange('A2').setValue(ideas);
}

/*
=====
* EXAMPLE 2 - A1 topic → A3:A7 (one idea per row)
*
=====
*/

function sheets_topicIdeasPerRow() {
  const sheet = getActiveSheetOrError_();
  const topic = sheet.getRange('A1').getValue();

  if (!topic) {
    sheet.getRange('A2').setValue('Please enter a topic in A1.');
```

return;

```
  }

  const response = callGemini_(
    'Give me 5 short bullet-point ideas (one per line) about: ' +
topic
  );

  const lines = response
    .split(/\r?\n/)
    .filter(line => line.trim() !== '')
    .map(line => [line.replace(/^-\+s*/, '').trim()]);
}

```

```

    if (lines.length === 0) {
        sheet.getRange('A3').setValue('No ideas generated.');
```

return;

```
    }

    sheet.getRange(3, 1, lines.length, 1).setValues(lines);
}

/*
=====
* EXAMPLE 3 - Explain formula in active cell
*
=====
*/

function sheets_explainFormula() {
    const sheet = getActiveSheetOrError_();
    const cell = sheet.getActiveCell();
    const formula = cell.getFormula();

    if (!formula) {
        SpreadsheetApp.getUi().alert('Active cell does not contain a
formula.');
```

return;

```
    }

    const explanation = callGemini_(
        'Explain what this Google Sheets formula does in simple
terms:\n\n' +
        formula
    );
    SpreadsheetApp.getUi().alert(explanation);
}

/*
=====
* EXAMPLE 4 - Translate A2:A → B2:B into chosen language
*
=====
*/

function sheets_translateColumn() {
    const sheet = getActiveSheetOrError_();
    const ui = SpreadsheetApp.getUi();

```

```

const langPrompt = ui.prompt(
  'Translation',
  'Translate to which language? (e.g., Spanish, French)',
  ui.ButtonSet.OK_CANCEL
);

if (langPrompt.getSelectedButton() !== ui.Button.OK) return;

const targetLang = langPrompt.getResponseText().trim();
if (!targetLang) {
  ui.alert('Please enter a valid language.');
```

return;

```

}

const values = sheet.getRange('A2:A').getValues();
const out = [];

values.forEach(row => {
  const text = row[0];
  if (!text) {
    out.push(['']);
    return;
  }
  const translated = callGemini_(
    'Translate the following text into ' + targetLang + ':\n\n' +
text
  );
  out.push([translated]);
});

sheet.getRange(2, 2, out.length, 1).setValues(out);
}

/*
=====
* EXAMPLE 5 – Extract TODOs from meeting notes in A1 → B2:B
*
=====
*/

function sheets_todosFromNotes() {
  const sheet = getActiveSheetOrError_();
  const notes = sheet.getRange('A1').getValue();

```

```

    if (!notes) {
      sheet.getRange('B1').setValue('Enter meeting notes in A1 first.');
```

```

    }

    const result = callGemini_(
      'From these meeting notes, extract a list of clear action items. '
+      'Return one item per line:\n\n' +
      notes
    );

    const lines = result
      .split(/\r?\n/)
      .filter(line => line.trim() !== '')
      .map(line => [line.replace(/^-\+\s*/, '').trim()]);

    if (lines.length === 0) {
      sheet.getRange('B2').setValue('No action items found.');
```

```

    }

    sheet.getRange(2, 2, lines.length, 1).setValues(lines);
  }

  /*
  =====
  * EXAMPLE 6 - Email draft from topic + recipient (A1, B1, C1 → D1)
  *
  =====
  */

  function sheets_emailDraft() {
    const sheet = getActiveSheetOrError_();
    const topic = sheet.getRange('A1').getValue();
    const recipient = sheet.getRange('B1').getValue();
    const context = sheet.getRange('C1').getValue();

    if (!topic || !recipient) {
      sheet
        .getRange('D1')
        .setValue('Please enter a topic in A1 and recipient name in
B1.');
```

```

    return;
  }

  const draft = callGemini_(
    'Write a polite email to ' +
      recipient +
      ' about: ' +
      topic +
      '.\n\nExtra context (optional):\n' +
      (context || '(none)')
  );

  sheet.getRange('D1').setValue(draft);
}

/*
=====
* EXAMPLE 7 - Meeting summary from A2:A10 → B1
*
=====
*/

function sheets_meetingSummary() {
  const sheet = getActiveSheetOnError_();
  const values = sheet.getRange('A2:A10').getValues();

  const notes = values
    .map(row => row[0])
    .filter(Boolean)
    .map(line => '- ' + line)
    .join('\n');

  if (!notes) {
    sheet.getRange('B1').setValue('Add meeting notes in A2:A10 first.');
```

```

    return;
  }

  const summary = callGemini_(
    'Summarize these meeting notes into a short paragraph and 3 bullet
points:\n\n' +
      notes
  );
}
```



```

    sheet.getRange('B1').setValue(summary);
}

/*
=====
* EXAMPLE 8 – Tutor answer from A1 question → B1
*
=====
*/

function sheets_tutorAnswer() {
    const sheet = getActiveSheetOrError_();
    const question = sheet.getRange('A1').getValue();

    if (!question) {
        sheet
            .getRange('B1')
            .setValue('Enter a question in A1, e.g. "What is a JavaScript
closure?"');
        return;
    }

    const answer = callGemini_(
        'You are a patient tutor. Answer this question step by step in
simple language:\n\n' +
        question
    );

    sheet.getRange('B1').setValue(answer);
}

/*
=====
* Simple test runner (non-destructive examples)
*
=====
*/

function test_allSheetsExamples() {
    // These are "safe" to run as long as A1 etc. have some content.
    sheets_topicIdeas();
    sheets_topicIdeasPerRow();
    // sheets_explainFormula(); // requires an active formula cell
    // sheets_translateColumn(); // interactive prompt, can run manually

```

```

sheets_todosFromNotes();
sheets_emailDraft();
sheets_meetingSummary();
sheets_tutorAnswer();
}

```

A1	Onboarding new JavaScript students						
	A	B	C	D	E	F	
1	Onboarding new Alex	Alex just enrolled in an online JavaScript bootcamp and is a bit nervous.					
2	Discussed welcome email sequence for new JS students						
3	Need a short orientation message for the LMS						
4	Decide on weekly Q&A time for live support						
5	Remind students about project-based learning focus						
6	Create a quick-start guide with 3 simple exercises						
7							

Below is a **helper script** that will:

- Create **3 sheets** (if they don't exist yet):
 - Ideas_Emails – for topic ideas, ideas per row, email drafts, TODOS
 - Translate_Summary – for translation + meeting summary
 - Tutor – for the tutor-style Q&A
- Pre-populate each sheet with **sensible example data**
- Work with the **functions you already have** (sheets_topicIdeas, sheets_meetingSummary, etc.), which use the **active sheet**

🧩 You just need to paste this into the **same project** as your Sheets Gemini code (where `getActiveSheetOnError_()` already exists).



Script to populate data for all exercises

```

/**
 * Populate sample data for all Gemini + Sheets examples.
 *
 * Creates (or reuses) three sheets:
 * - Ideas_Emails      → topic ideas, ideas per row, email draft,
TODOS
 * - Translate_Summary → translation + meeting summary
 * - Tutor              → tutor-style Q&A
 *
 * Then fills them with example values so you can run your functions
 * without manually typing data.
 */

```

```

function setupGeminiExampleData() {
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  if (!ss) {
    throw new Error('Open a Google Sheet and bind this script to it first.');
```

 }

 setupIdeasEmailsSheet_(ss);
 setupTranslateSummarySheet_(ss);
 setupTutorSheet_(ss);

 SpreadsheetApp.getUi().alert(
 'Sample data created.\n\n' +
 'Sheets:\n' +
 '- Ideas_Emails\n' +
 '- Translate_Summary\n' +
 '- Tutor\n\n' +
 'Switch to a sheet, then run the matching functions (e.g. sheets_topicIdeas).');
);
}

/**
 * Create or reset the Ideas_Emails sheet.
 *
 * Used by:
 * - sheets_topicIdeas()
 * - sheets_topicIdeasPerRow()
 * - sheets_todosFromNotes()
 * - sheets_emailDraft()
 */
function setupIdeasEmailsSheet_(ss) {
 const name = 'Ideas_Emails';
 const sheet = ss.getSheetByName(name) || ss.insertSheet(name);

 // Clear a safe range (don't nuke entire workbook)
 sheet.getRange('A1:D20').clearContent();

 // A1: topic (for ideas, ideas per row, TODOS)
 sheet.getRange('A1').setValue('Onboarding new JavaScript students');

 // B1: recipient name (for email draft)
 sheet.getRange('B1').setValue('Alex');

```

    // C1: extra context (for email draft)
    sheet.getRange('C1').setValue(
        'Alex just enrolled in an online JavaScript bootcamp and is a bit
nervous.'
    );

    // A2:A6: sample meeting notes (also work as "things to generate
    TODOS from")
    const meetingNotes = [
        ['Discussed welcome email sequence for new JS students'],
        ['Need a short orientation message for the LMS'],
        ['Decide on weekly Q&A time for live support'],
        ['Remind students about project-based learning focus'],
        ['Create a quick-start guide with 3 simple exercises']
    ];
    sheet.getRange(2, 1, meetingNotes.length,
1).setValues(meetingNotes);

    // Headers for convenience
    sheet.getRange('A1').setNote('Topic / General context');
    sheet.getRange('B1').setNote('Recipient name for email draft');
    sheet.getRange('C1').setNote('Extra context for email draft');
    sheet.getRange('A2').setNote('Meeting notes / content for TODOS');
}

/**
 * Create or reset the Translate_Summary sheet.
 *
 * Used by:
 * - sheets_translateColumn()
 * - sheets_meetingSummary()
 */
function setupTranslateSummarySheet_(ss) {
    const name = 'Translate_Summary';
    const sheet = ss.getSheetByName(name) || ss.insertSheet(name);

    sheet.getRange('A1:B20').clearContent();

    // A1: label / hint
    sheet.getRange('A1').setValue('Meeting notes & phrases (A2:A)');

    // A2:A7: text used BOTH as phrases to translate and as meeting
    notes
    const phrases = [

```

```

    ['Welcome to the JavaScript online bootcamp.'],
    ['Remember to review the DOM basics before the first project.'],
    ['Office hours are available every Wednesday afternoon.'],
    ['Ask questions early if you feel stuck on an exercise.'],
    ['Collaboration on projects is encouraged, but submit your own
code.'],
    ['Watch the intro video before starting Lesson 1.'],
  ];
  sheet.getRange(2, 1, phrases.length, 1).setValues(phrases);

  sheet.getRange('A1').setNote(
    'A2:A will be translated by sheets_translateColumn() and also
summarized by sheets_meetingSummary().'
  );
}

/**
 * Create or reset the Tutor sheet.
 *
 * Used by:
 * - sheets_tutorAnswer()
 */
function setupTutorSheet_(ss) {
  const name = 'Tutor';
  const sheet = ss.getSheetByName(name) || ss.insertSheet(name);

  sheet.getRange('A1:B10').clearContent();

  // A1: question for tutor-style answer
  sheet
    .getRange('A1')
    .setValue('What is a JavaScript closure and why is it useful?');

  sheet
    .getRange('A1')
    .setNote('Used by sheets_tutorAnswer() to generate a step-by-step
explanation.');
```

```

}

/**
 * Quick helper: jump between example sheets (optional).
 * Call this with:
 *   switchToSheet_("Ideas_Emails")
 */

```

```
function switchToSheet_(name) {
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const sheet = ss.getSheetByName(name);
  if (!sheet) {
    throw new Error('Sheet not found: ' + name);
  }
  ss.setActiveSheet(sheet);
}
```



How to use this with your existing functions

Assuming you already have the previous code (GEMINI, sheets_topicIdeas, etc.) in the same project:

1. **Run** `setupGeminiExampleData()` from the Script Editor.

- Authorize if prompted.
- It will create / reset:
 - Ideas_Emails
 - Translate_Summary
 - Tutor

2. For each sheet:

Sheet: Ideas_Emails

- Set this sheet active (click its tab).
- Run:
 - `sheets_topicIdeas()` → fills **A2** with ideas
 - `sheets_topicIdeasPerRow()` → fills **A3:A7** with one idea per row
 - `sheets_todosFromNotes()` → uses **A1 + A2:A...** to write todos into **B2:B...**
 - `sheets_emailDraft()` → uses **A1/B1/C1** to write an email into **D1**

3. **Sheet: Translate_Summary**

- Set this sheet active.

- Run:
 - `sheets_translateColumn()` → prompts for language, translates **A2:A** → **B2:B**
 - `sheets_meetingSummary()` → summarizes **A2:A10** into **B1**

4. **Sheet: Tutor**

- Set this sheet active.
- Run:
 - `sheets_tutorAnswer()` → reads **A1** (question), writes explanation into **B1**