

Google Docs + Gemini AI — Complete Guide for Beginners

Supercharge Google Docs with Gemini



Build AI-powered writing tools directly inside Google Docs using Apps Script

This guide walks you through **five AI tools** you can add to any Google Doc using Apps Script and Google's Gemini API.

Get more Apps Script Content at <https://basescripts.com/> by Laurence Svekis

You'll learn how each function works, how to use it, and how to customize it — even if you're new to Apps Script.

Overview of the Tools

This project includes **five AI-powered features** inside Google Docs:

1. **Summarize selected text**
2. **Rewrite selected text in simpler language**
3. **Generate an outline from a topic**
4. **Create multiple-choice quiz questions from selected text**
5. **Suggest a constructive writing comment**

All functions use a shared Gemini helper:

- `callGemini(prompt)` → sends a prompt to Gemini and returns text
 - Works with your own Gemini API key stored safely in Script Properties
-

Project Structure

Your Apps Script project includes:

- **SharedHelpers.gs** → Gemini API helper functions
- **DocsExamples.gs** → All 5 Google Docs AI tools

This guide explains only the Docs functions.

Setup Instructions

1. Open a Google Doc

Go to **Extensions** → **Apps Script**.

2. Create two files:

- `SharedHelpers.gs`
- `DocsExamples.gs`

3. Add your API key

In the Script editor:

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

Key: `GEMINI_API_KEY`

Value: your Gemini API key

4. Paste the code

Add `SharedHelpers` + `DocsExamples` code.

5. Save, authorize, and you're ready.

EXAMPLE 1 — Summarize Selected Text

✓ What it does

Select text in your Google Doc → run `docs_summarizeSelection()` → Gemini generates a **short summary**.

✓ Use cases

- Meeting notes
- Research papers
- Lesson content
- Long emails
- Blog drafts
- Academic reading

✓ Code

```
function docs_summarizeSelection() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

  if (!sel) {
    DocumentApp.getUi().alert('Select some text first.');
```

```
    return;
  }

  let text = '';
  sel.getRangeElements().forEach(el => {
    const element = el.getElement();
    if (element.editAsText) {
      text += element.asText().getText() + '\n';
    }
  });

  if (!text.trim()) {
    DocumentApp.getUi().alert('Selected text is empty.');
```

```
    return;
  }

  const summary = callGemini('Summarize this text in a short
paragraph:\n\n' + text);
  DocumentApp.getUi().alert(summary);
}
```

✓ How it works

- Gets the current document

- Retrieves the user's text selection
- Extracts only editable text nodes
- Passes the text to Gemini
- Shows the summary in a popup



Exercise

Select 2–3 paragraphs of text and run the function.

Try different writing styles: emails, scripts, policy documents, or lecture notes.

EXAMPLE 2 — Rewrite Text in Simpler Language

✓ What it does

Rewrites selected text at a **Grade 6 reading level**.

✓ Use cases

- Education
- Accessibility
- Simplifying instructions
- Making content easier to understand

✓ Code

```
function docs_rewriteSimpler() {  
  const doc = DocumentApp.getActiveDocument();  
  const sel = doc.getSelection();
```

```

if (!sel) {
  DocumentApp.getUi().alert('Select some text first.');
```

return;

```

}

let text = '';
sel.getRangeElements().forEach(el => {
  const element = el.getElement();
  if (element.editAsText) {
    text += element.asText().getText() + '\n';
  }
});

if (!text.trim()) {
  DocumentApp.getUi().alert('Selected text is empty.');
```

return;

```

}

const simpler = callGemini(
  'Rewrite the following text at a Grade 6 reading level:\n\n' +
text
);
DocumentApp.getUi().alert(simpler);
}

```

✓ How it works

Same logic as Example 1 — but Gemini receives a rewriting instruction.



Exercise

Try rewriting:

- A legal paragraph
- A technical explanation
- A blog paragraph
- A definition

Compare the before/after results.

EXAMPLE 3 — Generate Outline From Topic

✓ What it does

Asks the user for a topic → generates a **structured bullet-point outline** → inserts it into the Doc.

✓ Use cases

- Lesson planning
- Blog post drafting
- Speech preparation
- Planning reports or presentations
- Breaking down large topics

✓ Code

```
function docs_generateOutlineFromTopic() {
  const ui = DocumentApp.getUi();
  const response = ui.prompt(
    'Outline Generator',
    'Enter a topic:',
    ui.ButtonSet.OK_CANCEL
  );

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

  const topic = response.getResponseText().trim();
  if (!topic) {
    ui.alert('Please enter a topic.');
```

```
const outline = callGemini(  
  'Create a short bullet-point outline for this topic:\n\n' + topic  
);  
  
const doc = DocumentApp.getActiveDocument();  
const body = doc.getBody();  
body.appendParagraph('\nOutline for: ' + topic).setBold(true);  
body.appendParagraph(outline);  
}
```

✓ How it works

- Shows a prompt dialog
- Extracts the topic
- Calls Gemini to produce a list
- Inserts the outline into your document



Exercise

Generate outlines for:

- “Intro to JavaScript”
- “Climate Change”
- “Beginner’s Guide to Investing”
- “Marketing Plan for 2025”



EXAMPLE 4 — Create Quiz Questions From Text

✓ What it does

Select text → creates **5 multiple-choice questions** with 4 options each, including the correct answer.

✓ Use cases

- Teachers
- Online courses
- Corporate training
- Study guides
- Educational content creation

✓ Code

```
function docs_quizFromSelection() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

  if (!sel) {
    DocumentApp.getUi().alert('Select some text first.');
```

```
    return;
  }

  let text = '';
  sel.getRangeElements().forEach(el => {
    const element = el.getElement();
    if (element.editAsText) {
      text += element.asText().getText() + '\n';
    }
  });

  if (!text.trim()) {
    DocumentApp.getUi().alert('Selected text is empty.');
```

```
    return;
  }

  const quiz = callGemini(
```

```

    'Based on the text below, create 5 multiple choice questions with
4 options each and mark the correct answer:\n\n' +
    text
);

const body = doc.getBody();
body.appendParagraph('\nGenerated Quiz:').setBold(true);
body.appendParagraph(quiz);
}

```

✓ How it works

- Extracts selected text
- Passes it to Gemini with quiz-generation instructions
- Inserts the quiz into the Doc



Exercise

Select a Wikipedia paragraph and generate a quiz.
Try with chapters from textbooks, training documents, or blog posts.

EXAMPLE 5 — Suggest a Constructive Comment

✓ What it does

Reads selected text → suggests an **editor-style writing improvement**.

✓ Use cases

- Editing drafts
- Peer review
- Improving clarity
- Coaching writers
- Giving feedback

✓ Code

```
function docs_suggestComment() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

  if (!sel) {
    DocumentApp.getUi().alert('Select some text first.');
```

```
    return;
  }

  let text = '';
  sel.getRangeElements().forEach(el => {
    const element = el.getElement();
    if (element.editAsText) {
      text += element.asText().getText() + '\n';
    }
  });

  if (!text.trim()) {
    DocumentApp.getUi().alert('Selected text is empty.');
```

```
    return;
  }

  const comment = callGemini(
    'You are a helpful writing coach. Suggest one constructive comment
to improve this text:\n\n' +
    text
  );

  DocumentApp.getUi().alert('Suggested comment:\n\n' + comment);
}
```

✓ How it works

Prompts Gemini to behave like a writing coach and evaluate the selected text.

Exercise

Try selecting:

- A paragraph of your own writing
 - A blog intro
 - A product description
 - A social media post
-

How All Functions Work Behind the Scenes

Step 1 — Detect the user's selection

Google Docs scripts can detect selected elements, but not styles or layout.

Step 2 — Combine the text into a single string

Line separation helps Gemini process the content.

Step 3 — Generate the prompt

Each function uses a clear natural-language instruction.

Step 4 — Send the prompt to Gemini

`callGemini(prompt)` does all the heavy lifting.

Step 5 — Display or insert the result

Alerts are used for display, and the document body for insertions.



How to Run These Functions

Method 1 — Direct from editor

In Apps Script:

- Select a function (e.g., docs_summarizeSelection)
- Click ► Run

Method 2 — Add a custom menu (optional)

I can generate this if you want:

- ✓ “AI Tools” menu
 - ✓ Buttons for each feature
 - ✓ No coding knowledge required for users
-



Final Thoughts

These five examples give you a full foundation for building advanced Gemini tools inside Google Docs. From rewriting to summarizing, generating outlines, creating quizzes, or giving feedback — this toolkit turns Docs into a **smart AI writing assistant**.