

Google Apps Script for Google Docs: Comprehensive Guide



What is Google Apps Script?	2
Benefits:	2
How to Access Google Apps Script for Docs	2
Key Google Docs Apps Script Methods	2
Basic Example: Insert Text in Google Docs	2
Working with Formatting	3
Example: Formatting Text	3
Creating and Styling Tables	3
Example: Add a Table	3
Inserting Images	4
Example: Add an Image	4
Automating Document Creation	4
Example: Create a New Document	4
Exercises	4
Exercise 1: Add and Format a Heading	4
Solution:	5

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

Exercise 2: Generate a Custom Table	5
Solution:	5
Exercise 3: Add a Footer	5
Solution:	5
Multiple-Choice Questions	5
Question 1:	6
Question 2:	6
Question 3:	6
Advanced Example: Automate Invoice Creation	6

Google Apps Script is a cloud-based JavaScript platform that allows you to automate tasks and extend the functionality of Google Docs. This guide covers the basics, key methods, detailed examples, and exercises to master Google Apps Script with Google Docs.

What is Google Apps Script?

Google Apps Script is a JavaScript-based scripting language for automating Google Workspace applications such as Google Docs, Sheets, and Drive.

Benefits:

- Automates repetitive tasks.
- Enhances Google Docs with custom functionality.
- Integrates Google Workspace apps.

How to Access Google Apps Script for Docs

1. Open a Google Document.
2. Click **Extensions > Apps Script**.
3. In the Apps Script editor, write your code and save.

Key Google Docs Apps Script Methods

1. **DocumentApp**: Represents the Google Document.
2. **Body**: Represents the main content of the document.
3. **Paragraph**: Represents paragraphs in the body.
4. **Text**: Manipulates the text within a document.

Basic Example: Insert Text in Google Docs

```
function insertText() {
```

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

```
    const doc = DocumentApp.getActiveDocument(); // Open the current
Google Doc
    const body = doc.getBody(); // Get the body of the document
    body.appendParagraph("Hello, Google Apps Script!");
}
```

Explanation:

- `DocumentApp.getActiveDocument()`: Opens the active document.
- `body.appendParagraph("...")`: Adds a new paragraph at the end of the document.

Working with Formatting

Example: Formatting Text

```
function formatText() {
    const doc = DocumentApp.getActiveDocument();
    const body = doc.getBody();
    const paragraph = body.appendParagraph("This is bold and red
text.");
    paragraph.setBold(true).setForegroundColor("red");
}
```

Explanation:

- `setBold(true)`: Makes the text bold.
- `setForegroundColor("red")`: Changes the text color to red.

Creating and Styling Tables

Example: Add a Table

```
function createTable() {
    const doc = DocumentApp.getActiveDocument();
    const body = doc.getBody();
    const table = body.appendTable([
        ["Name", "Age", "Email"],
        ["Alice", "25", "alice@example.com"],
        ["Bob", "30", "bob@example.com"]
    ]);
    table.setBorderWidth(2);
}
```

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

Explanation:

- `appendTable([...])`: Adds a table to the document.
- `setBorderWidth(2)`: Sets the border width for the table.

Inserting Images

Example: Add an Image

```
function insertImage() {
  const doc = DocumentApp.getActiveDocument();
  const body = doc.getBody();
  const imageUrl = "https://via.placeholder.com/150";
  body.appendImage(UrlFetchApp.fetch(imageUrl).getBlob());
}
```

Explanation:

- `appendImage()`: Inserts an image into the document.
- `UrlFetchApp.fetch(imageUrl).getBlob()`: Fetches the image as a blob object.

Automating Document Creation

Example: Create a New Document

```
function createNewDoc() {
  const newDoc = DocumentApp.create("My New Document");
  const body = newDoc.getBody();
  body.appendParagraph("This is a new document created via Apps
Script.");
}
```

Explanation:

- `DocumentApp.create("...")`: Creates a new Google Doc.
- `appendParagraph()`: Adds a paragraph to the new document.

Exercises

Exercise 1: Add and Format a Heading

Write a script to insert a heading at the top of a Google Doc, make it bold, and change its font size to 24.

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

Solution:

```
function addHeading() {
  const doc = DocumentApp.getActiveDocument();
  const body = doc.getBody();
  const heading = body.insertParagraph(0, "Document Heading");
  heading.setHeading(DocumentApp.ParagraphHeading.HEADING1);
  heading.setFontSize(24).setBold(true);
}
```

Exercise 2: Generate a Custom Table

Write a script to create a table with 3 rows and 4 columns. Populate it with sample data.

Solution:

```
function customTable() {
  const doc = DocumentApp.getActiveDocument();
  const body = doc.getBody();
  const table = body.appendTable();
  for (let i = 0; i < 3; i++) {
    const row = table.appendTableRow();
    for (let j = 0; j < 4; j++) {
      row.appendTableCell(`Row ${i + 1}, Col ${j + 1}`);
    }
  }
}
```

Exercise 3: Add a Footer

Write a script to add a footer with the current date to a Google Doc.

Solution:

```
function addFooter() {
  const doc = DocumentApp.getActiveDocument();
  const footer = doc.addFooter();
  const date = new Date().toLocaleDateString();
  footer.appendParagraph(`Document generated on: ${date}`);
}
```

Multiple-Choice Questions

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

Question 1:

Which method is used to fetch the active Google Document?

1. `DocumentApp.open()`
2. `DocumentApp.getActiveDocument()`
3. `Document.openActive()`
4. `DocumentApp.currentDocument()`

Answer: 2. `DocumentApp.getActiveDocument()`

Question 2:

What does `appendParagraph()` do in Google Apps Script?

1. Inserts a paragraph at the top of the document.
2. Replaces the existing paragraph.
3. Appends a paragraph to the document body.
4. Deletes a paragraph from the document.

Answer: 3. Appends a paragraph to the document body.

Question 3:

Which of the following is NOT a valid method for formatting text in Google Apps Script?

1. `setBold()`
2. `setUnderline()`
3. `setTextAlignment()`
4. `setImageSize()`

Answer: 4. `setImageSize()`

Explanation: `setImageSize()` is not used for text formatting.

Advanced Example: Automate Invoice Creation

```
function generateInvoice() {  
  const doc = DocumentApp.create("Invoice");  
  const body = doc.getBody();
```

```
  body.appendParagraph("Invoice").setHeading(DocumentApp.ParagraphHeading.HEADING1);
```

```
  body.appendParagraph(`Date: ${new Date().toLocaleDateString()}`);
```

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

```
body.appendParagraph("Customer Name: John Doe");
const table = body.appendTable([
  ["Item", "Quantity", "Price", "Total"],
  ["Widget A", "2", "$10", "$20"],
  ["Widget B", "5", "$8", "$40"],
  ["", "", "Total:", "$60"]
]);
table.setBorderWidth(1);
}
```

Explanation:

- Automates the creation of an invoice document with a table summarizing items.
- Adds headings and formatted content.