

Apps Script Code Exercises

🌟 Master Google Docs with Google Apps Script! 🌟



🌟 Master Google Docs with Google Apps Script! 🌟	1
Exercise 1: Create a New Google Document	2
Exercise 2: Add Text to a Document	2
Exercise 3: Formatting Text	3
Exercise 4: Inserting a Table	4
Exercise 5: Adding Headers and Footers	4
Exercise 6: Working with Images	5
Exercise 7: Creating a List	6
Exercise 8: Working with Styles	6
Exercise 9: Linking Text	7
Exercise 10: Collaborative Editing Features	7

🌟 Master Google Docs with Google Apps Script! 🌟

10 hands-on coding exercises designed to boost your skills in using Google Apps Script's Document Service (DocumentApp). These exercises cover everything from

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

creating and formatting documents to advanced features like inserting images and managing collaborative editing tools.



Whether you're a beginner or a seasoned pro, these exercises will enhance your ability to automate and manipulate Google Docs, streamlining your workflow and boosting productivity.



Dive into these exercises and transform the way you use Google Docs!

Exercise 1: Create a New Google Document

Objective: Learn to create a new Google Document using DocumentApp.

Explanation: This exercise teaches how to create a new document in Google Drive using Google Apps Script.

Code:

```
function createDocument() {  
  var doc = DocumentApp.create('New Document');  
  Logger.log(doc.getUrl());  
}
```

Exercise 2: Add Text to a Document

Objective: Learn to add text to a Google Document.

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Explanation: This exercise focuses on how to insert text into a document, which is a fundamental skill.

Code:

```
function addTextToDocument() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  body.appendParagraph('Hello, World!');  
}
```

Exercise 3: Formatting Text

Objective: Understand how to format text in a document.

Explanation: This exercise will show you how to change font size, color, and style of the text.

Code:

```
function formatText() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  var paragraph = body.appendParagraph('Formatted Text');  
  paragraph.setFontSize(12);  
  paragraph.setForegroundColor('red');  
  paragraph.setBold(true);  
}
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
}
```

Exercise 4: Inserting a Table

Objective: Learn to insert a table into a document.

Explanation: Tables are essential for organizing data. This exercise teaches how to create and populate a table.

Code:

```
function insertTable() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  var table = body.appendTable();  
  var row = table.appendTableRow();  
  row.appendTableCell('Cell 1');  
  row.appendTableCell('Cell 2');  
}
```

Exercise 5: Adding Headers and Footers

Objective: Understand how to add headers and footers.

Explanation: This exercise focuses on adding headers and footers, which is important for document formatting.

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Code:

```
function addHeadersFooters() {  
  var doc = DocumentApp.getActiveDocument();  
  var header = doc.addHeader();  
  var footer = doc.addFooter();  
  header.appendParagraph('Header Text');  
  footer.appendParagraph('Footer Text');  
}
```

Exercise 6: Working with Images

Objective: Learn to insert and manipulate images in a document.

Explanation: This exercise will demonstrate how to add images and modify their properties.

Code:

```
function insertImage() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  var imageUrl = 'https://example.com/image.png'; // Replace with actual image  
  URL  
  var image = UrlFetchApp.fetch(imageUrl).getBlob();  
  body.appendImage(image);  
}
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Exercise 7: Creating a List

Objective: Learn to create bulleted and numbered lists.

Explanation: Lists are crucial for organization. This exercise teaches how to create and format lists.

Code:

```
function createList() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  var items = ['Item 1', 'Item 2', 'Item 3'];  
  body.appendListItem(items[0]).setGlyphType(DocumentApp.GlyphType.BULLETS);  
  body.appendListItem(items[1]).setGlyphType(DocumentApp.GlyphType.BULLETS);  
  body.appendListItem(items[2]).setNumbered(true);  
}
```

Exercise 8: Working with Styles

Objective: Understand how to apply and customize styles in a document.

Explanation: This exercise will show you how to use predefined styles and create custom styles.

Code:

```
function applyStyles() {
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
var doc = DocumentApp.getActiveDocument();
var body = doc.getBody();
var paragraph = body.appendParagraph('Styled Text');
paragraph.setHeading(DocumentApp.ParagraphHeading.Heading1);
}
```

Exercise 9: Linking Text

Objective: Learn how to add hyperlinks to text.

Explanation: This exercise focuses on how to link text to URLs, an important feature for interactive documents.

Code:

```
function addHyperlink() {
  var doc = DocumentApp.getActiveDocument();
  var body = doc.getBody();
  var text = body.appendParagraph('Google Website');
  text.setLinkUrl('https://www.google.com');
}
```

Exercise 10: Collaborative Editing Features

Objective: Explore features like comments and suggestions.

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Explanation: Learn to programmatically add comments and suggestions, which are crucial for collaborative editing.

Code:

```
function addComment() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  var text = body.appendParagraph('Text to comment on');  
  var rangeBuilder = doc.newRange();  
  rangeBuilder.addElement(text);  
  doc.addComment(rangeBuilder.build(), 'This is a comment.');
```

```
}
```