





Apps Script **Code Exercises**

 Enhance Your
Google Sheets Skills
with Apps Script! 



 Enhance Your Google Sheets Skills with Apps Script! 	1
Exercise 1: Create a New Spreadsheet	2
Exercise 2: Write Data to a Spreadsheet	2
Exercise 3: Read Data from a Spreadsheet	3
Exercise 4: Formatting Cells	3
Exercise 5: Adding Multiple Rows of Data	4
Exercise 6: Creating a Custom Function	4
Exercise 7: Fetching Data from External APIs	5
Exercise 8: Looping Through Rows	6
Exercise 9: Using Triggers	6
Exercise 10: Creating Charts	7

 Enhance Your Google Sheets Skills with Apps Script!


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

🔥 10 interactive coding exercises to master Google Apps Script, specifically the Spreadsheet Service (SpreadsheetApp). These exercises range from basic to advanced levels, ensuring a comprehensive learning journey.

✅ From creating and formatting spreadsheets to automating tasks with triggers, you'll learn it all! Plus, delve into fetching data from external APIs and creating dynamic charts.

Exercise 1: Create a New Spreadsheet

Objective: Learn how to create a new spreadsheet using SpreadsheetApp.

Explanation: This exercise will teach you how to create a new spreadsheet. You'll learn to use the create method of SpreadsheetApp.

Code:

```
function createSpreadsheet() {  
  var spreadsheet = SpreadsheetApp.create("New Spreadsheet");  
  Logger.log(spreadsheet.getUrl());  
}
```

Exercise 2: Write Data to a Spreadsheet

Objective: Learn how to write data to specific cells.

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 writing data to a spreadsheet. You'll use methods like `getActiveSheet` and `getRange`.

Code:

```
function writeData() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  sheet.getRange("A1").setValue("Hello World!");  
}
```

Exercise 3: Read Data from a Spreadsheet

Objective: Learn how to read data from a spreadsheet.

Explanation: Here, you'll learn to read data from a cell. This is crucial for processing and analyzing spreadsheet data.

Code:

```
function readData() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var data = sheet.getRange("A1").getValue();  
  Logger.log(data);  
}
```

Exercise 4: Formatting Cells

Objective: Understand how to format cells in a spreadsheet.

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

Explanation: This exercise will show you how to change the font size, color, and cell background.

Code:

```
function formatCells() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getRange("A1:A10");  
  range.setFontSize(12);  
  range.setBackground("yellow");  
  range.setFontColor("red");  
}
```

Exercise 5: Adding Multiple Rows of Data

Objective: Learn to add multiple rows of data to a spreadsheet.

Explanation: You'll learn how to use arrays and the `setValues` method to add multiple rows of data efficiently.

Code:

```
function addMultipleRows() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var values = [{"Row 1", 1}, {"Row 2", 2}, {"Row 3", 3}];  
  sheet.getRange("A1:B3").setValues(values);  
}
```

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

```
}
```

Exercise 6: Creating a Custom Function

Objective: Create a custom function in Google Sheets using Apps Script.

Explanation: This teaches you to create a function that can be used directly in your Google Sheets as a formula.

Code:

```
function doubleNumber(number) {  
  return number * 2;  
}
```

// Usage in Google Sheets: =doubleNumber(A1)

Exercise 7: Fetching Data from External APIs

Objective: Learn to fetch data from external APIs and populate it in a spreadsheet.

Explanation: This advanced exercise will help you understand how to connect with external data sources.

Code:

```
function fetchData() {  
  var response = UrlFetchApp.fetch("https://api.example.com/data");  
  var data = JSON.parse(response.getContentText());  
}
```

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

```
// Process and write data to the sheet  
}
```

Exercise 8: Looping Through Rows

Objective: Understand how to loop through rows in a spreadsheet.

Explanation: You'll learn to use loops to process or analyze data in a spreadsheet.

Code:

```
function loopThroughRows() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var data = sheet.getDataRange().getValues();  
  
  for (var i = 0; i < data.length; i++) {  
    var row = data[i];  
    // Process each row  
  }  
}
```

Exercise 9: Using Triggers

Objective: Learn to use Apps Script triggers like onEdit.

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

Explanation: Triggers allow scripts to run automatically. You'll create a simple onEdit trigger.

Code:

```
function onEdit(e) {  
  var range = e.range;  
  range.setBackground("green");  
}
```

Exercise 10: Creating Charts

Objective: Learn to create charts in a spreadsheet using Apps Script.

Explanation: This will teach you to programmatically create charts in your spreadsheet.

Code:

```
function createChart() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var chart = sheet.newChart()  
    .setChartType(Charts.ChartType.LINE)  
    .addRange(sheet.getRange("A1:B10"))  
    .setPosition(4, 8, 0, 0)  
    .build();  
}
```

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

```
sheet.insertChart(chart);  
}
```