Google Apps Script Coding Exercises

Apps Script
LEARN TO CODE
Learn EXERCISE

Laurence Svekis

Google Apps Script Exercises

Exercise 1: Sending Email

Task: Write a script that sends an email to a specific recipient with a predefined subject and body.

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses
https://basescripts.com/

Code:

```
function sendEmail() {

  var recipient = 'example@email.com';

  var subject = 'Greetings from Google Apps Script';

  var body = 'This is a test email sent using Google Apps Script!';



  GmailApp.sendEmail(recipient, subject, body);

}
```

Solution:

Run the sendEmail() function, and it will send the specified email.

Exercise 2: Spreadsheet Data Retrieval

Task: Fetch data from a Google Spreadsheet and log it.

Code:

```javascript
function retrieveDataFromSheet() {

  var spreadsheet =
    SpreadsheetApp.openById('your_spreadsheet_id');

  var sheet = spreadsheet.getSheetByName('Sheet1');

  var data = sheet.getDataRange().getValues();



  Logger.log(data);

}
```

Solution:

Replace 'your_spreadsheet_id' with the actual ID and run
    retrieveDataFromSheet() to log the data.

Exercise 3: Create a Draft Email

Task: Write a script that creates a draft email with predefined
    content.

Code:

```
function createDraftEmail() {

  var recipient = 'example@email.com';

  var subject = 'Draft Email';

  var body = 'This is a draft email created using Google Apps
    Script!';



  GmailApp.createDraft(recipient, subject, body);

}
```

Solution:

Run createDraftEmail() to create a draft with the specified content.

Exercise 4: Manipulate Data in Spreadsheet

Task: Update a Google Spreadsheet by adding a new row of data.

Code:

```
function addDataToSheet() {
```

```
  var spreadsheet =
    SpreadsheetApp.openById('your_spreadsheet_id');

  var sheet = spreadsheet.getSheetByName('Sheet1');

  var newData = ['John Doe', 'john.doe@email.com', 'Developer'];

    sheet.appendRow(newData);

}
```

Solution:

Replace 'your_spreadsheet_id' with the actual ID and run
    addDataToSheet() to add a new row.

Exercise 5: Mark Thread as Read

Task: Write a script that marks the first thread in the inbox as read.

Code:

```
function markThreadAsRead() {

  var thread = GmailApp.getInboxThreads(0, 1)[0];
```

```
  thread.markRead();

}
```

Solution:

Run markThreadAsRead() to mark the first thread in the inbox as read.

Exercise 6: Working with Dates

Task: Log the current date and time using Google Apps Script.

Code:

```
function logCurrentDate() {

  var currentDate = new Date();

  Logger.log('Current Date: ' + currentDate);

}
```

Solution:

Run logCurrentDate() to log the current date and time.

Exercise 7: Delete Draft by ID

Task: Write a script that deletes a draft by its ID.

Code:

```
function deleteDraftById(draftId) {

  var draft = GmailApp.getDraftById(draftId);

  draft.deleteDraft();

}
```

Solution:

Provide a valid draft ID as an argument and run
    deleteDraftById('your_draft_id') to delete the draft.

Exercise 8: Create a Menu in Google Sheets

Task: Create a custom menu in Google Sheets that runs a specific function.

Code:

```
function onOpen() {

  var ui = SpreadsheetApp.getUi();

  ui.createMenu('Custom Menu')

    .addItem('Run Function', 'yourFunction')

    .addToUi();

}



function yourFunction() {

  // Your function code here

}
```

Solution:

Refresh the Google Sheet, and you'll find a "Custom Menu" with a "Run Function" option.

Exercise 9: Accessing Cell Values

Task: Fetch the value of a specific cell in a Google Spreadsheet.

Code:

```
function getCellValue() {

  var spreadsheet =
    SpreadsheetApp.openById('your_spreadsheet_id');

  var sheet = spreadsheet.getSheetByName('Sheet1');

  var value = sheet.getRange('A1').getValue();



  Logger.log('Cell A1 Value: ' + value);
```

}

Solution:

Replace 'your_spreadsheet_id' with the actual ID and run getCellValue() to log the value of cell A1.

Exercise 10: Create a Data Table

Task: Generate a basic data table in the Logger.

Code:

```
function createDataTable() {

  var data = [

    ['Name', 'Age', 'Occupation'],

    ['John Doe', 30, 'Engineer'],

    ['Jane Smith', 25, 'Designer']

  ];
```

```javascript
  Logger.log('Data Table:');

  for (var i = 0; i < data.length; i++) {

    Logger.log(data[i].join('\t'));

  }

}
```

Solution:

Run createDataTable() to log a basic data table in the Logger.