



150 CODE Examples

Google Apps Script

Writing Data to a Google Sheet	4
Reading Data from a Google Sheet	5
Creating a New Google Doc	5
Sending an Email with GmailApp	5
Adding an Event to Google Calendar	5
Fetching Data from an External API	5
Creating a Custom Menu in Google Sheets	5
Using Google Sheets as a Database	6
Updating Google Sheet Data Based on Conditions	6
Formatting Cells in Google Sheets	6
Removing Duplicate Rows in Google Sheets	6
Inserting a Chart in Google Sheets	7
Convert Spreadsheet Data to JSON	7
Batch Update Google Sheet Cells	7
Auto-Resize Columns in Google Sheets	8
Find and Replace Text in Google Docs	8
List All Files in Google Drive Folder	8
Create a New Google Drive Folder	8
Send an Email with an Attachment from Google Drive	8
Add Custom Footer to Google Doc	9
Archiving Emails from Gmail	9
Create a Document Copy with a New Name	9
Inserting a New Row in Google Sheets After the Last Row with Data	9
Retrieve and Log the Names of All Spreadsheets in Google Drive	9
Add a Custom Status Column to Google Sheets Based on Another Column's Values	10
Automatically Color Code Rows in Google Sheets Based on Cell Value	10
Create and Email a PDF from a Google Sheet	10
Update Google Calendar Events Based on Google Sheets Data	11
Export Google Sheets Data to a New Google Doc	11
Automatically Backup Google Sheets to Another Spreadsheet	12

Setting Validation Rules in Google Sheets	12
Creating a Google Calendar Event with Guests	12
Appending a Row to a Google Sheet from Form Submission	12
Automatically Formatting New Rows in Google Sheets	13
Fetching and Logging Weather Data	13
Bulk Removing Rows Based on Condition in Google Sheets	13
Sending Customized Email Based on Google Sheets Data	13
Sync Google Sheets Data with Google Calendar	14
Generating a Report from Google Sheets and Emailing as a PDF	14
Removing Duplicate Entries in Google Sheets	14
Splitting a String into Multiple Cells in Google Sheets	15
Merging Multiple Google Sheets into One Sheet	15
Highlighting Cells Based on Expiry Date	16
Automatically Add a Timestamp to a Row on Edit	16
Convert Google Sheet Rows to JSON Objects	16
Creating a Function to Clear Specific Cells in Google Sheets	17
Function to Count Non-Empty Cells in a Specific Column	17
Sending a Daily Digest Email from a Google Sheet	17
Auto-Resizing Row Height in Google Sheets Based on Content	17
Function to Replace Text in a Google Doc	18
Batch Update Cell Background Color in Google Sheets	18
Create a List of Hyperlinks in Google Sheets	18
Summarize Data from Multiple Sheets into a Master Sheet	18
Convert Column Index to Letter in Google Sheets	19
Automate Email Sending Based on Cell Value Changes	19
Extract Email Addresses from Text	19
Generate a Random Password Generator in Google Sheets	20
Delete Rows Based on a Condition in Google Sheets	20
Create a Table of Contents for a Google Doc	20
Updating Google Calendar Event Descriptions in Bulk	20
Highlighting Weekends in Google Sheets	21
Inserting 'Last Updated' Timestamp in Google Sheets Footer	21
Copy Google Sheet Data to Another Spreadsheet	21
Clear Contents from Multiple Ranges in Google Sheets	22
Create Multiple Google Calendar Events from Google Sheets Data	22
Validate Email Format in Google Sheets Using Regex	22
Automatically Resize Images in Google Docs	23
Convert Google Sheets Range to HTML Table	23
Automatically Sort Data in Google Sheets When a Cell Changes	23
Fetch Latest Cryptocurrency Prices and Update Google Sheets	24
Compare Two Columns and Highlight Mismatches in Google Sheets	24

Automatically Hide Rows with Zero Values in Google Sheets	25
Send a Scheduled Email Reminder from Google Sheets Data	25
Export Specific Google Sheets Data to a New Sheet	25
Validate Data Entry as Email Format in Google Sheets	25
Log Changes in Google Sheets to a Separate Sheet	26
Generate QR Codes in Google Sheets	26
Summarize Google Forms Responses in Google Sheets	26
Auto-Generate Document Based on Google Sheets Data	27
Create and Format a New Google Sheet from Script	27
Change Google Sheets Tab Color Based on Cell Value	27
Automatically Number Rows in Google Sheets	28
Remove Empty Rows from the End of Google Sheets	28
Convert Selected Range to Uppercase in Google Sheets	28
Sync Google Sheets Data with a Secondary Sheet	29
Generate a List of All Hyperlinks in Google Sheets	29
Create Google Calendar Events from Selected Range in Google Sheets	29
Highlight Duplicate Entries in Google Sheets	30
Automatically Format New Google Sheets Entries as Currency	30
Compare Two Google Sheets for Differences	30
Append Google Form Responses to a Google Doc	31
Clear Formatting in a Selected Range in Google Sheets	31
Insert Current Date and Time in a Cell on Edit	31
Automatically Convert Incoming Emails to Google Docs	32
Create a Pie Chart in Google Sheets from Data	32
Send a Custom Slack Message from a Google Sheet	32
Update a Cell Based on Changes in Another Cell	32
Batch Create Google Calendar Events from Spreadsheet Data	33
Color Code Cells Based on Expiry Date in Google Sheets	33
Auto-Delete Rows Based on Cell Content in Google Sheets	34
Convert All Formulas to Values in Google Sheets	34
Auto-Generate and Email a Google Sheets Report as PDF	34
Set Data Validation for a Range Based on Another Column's Values	35
Highlight Cells in Google Sheets That Contain Formulas	35
Automatically Add Google Calendar Events to a Google Sheet	35
Create a Google Sheets Log for Failed Email Attempts	35
Batch Clear Multiple Named Ranges in Google Sheets	36
Merge Multiple Google Docs into One Document	36
Auto-Format Negative Numbers in Red in Google Sheets	36
Extract and Summarize Checked Items from Google Forms Checkbox Questions	37
Automate Data Backups in Google Sheets	37
Track Cell Edits and Timestamps in a Separate Google Sheets Tab	37

Summarize Selected Google Sheets Data in an Email	38
Find and Replace Text in Multiple Google Docs	38
Create a Dropdown List from Range in Google Sheets	38
Automatically Move Google Sheets Rows to Another Tab Based on Cell Value	39
Generate and Insert a Google Sheets Chart into a Google Slide	39
Extract All Email Addresses from a Gmail Thread	39
Auto-Create a Monthly Calendar in Google Sheets	40
Monitor and Log Google Sheets Cell Changes Over Time	40
Automatically Color-code Google Calendar Events Based on Keywords	40
Create a Directory of Files and Folders in Google Drive	41
Export Google Sheets Charts as Images	41
Sync Google Sheets Data with Google Contacts	41
Auto-Sort Google Sheets When a New Row is Added	42
Remove Duplicate Rows Based on a Specific Column in Google Sheets	42
Insert Current Weather Information into Google Sheets	42
Convert Google Sheet Rows to Individual Text Files	43
Monitor Google Sheets Cell Value and Send Alert if Changed	43
Create a Checklist in Google Sheets with Checkboxes	43
Auto-generate Google Calendar Events from Task List in Google Sheets	44
Highlight Overdue Tasks in Google Sheets	44
Convert Sheet Data to JSON and Log It	44
Batch Resize Images in Google Drive	45
Insert Formatted Date and Time in a Cell	45
Validate Email Addresses in a Column	45
Automatically Close Google Form Submissions After a Certain Number of Responses	45
Create a Progress Tracker in Google Sheets	46
Generate Unique ID for Each Google Sheets Row	46
Extract Specific Data from Gmail Messages to Google Sheets	47
Automatically Update Google Sheets with Stock Prices	47
Automatically Archive Gmail Messages in Google Sheets	47
Automatically Generate Google Calendar Events from Google Forms Responses	48
Automatically Send Email Notifications for Google Forms Responses	48
Automatically Create Google Calendar Events from Google Sheets Data	48
Automatically Track Changes to Google Sheets with Revision History	49
Automatically Generate QR Codes from URLs in Google Sheets	49
Automatically Generate Google Docs from Google Sheets Data	49
Automatically Translate Google Sheets Data	50
Automatically Generate Google Slides Presentations from Google Sheets Data	50

Writing Data to a Google Sheet

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

Reading Data from a Google Sheet

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

Creating a New Google Doc

```
function createGoogleDoc() {  
  var doc = DocumentApp.create('New Document');  
  doc.getBody().appendParagraph('This is a new document.');
```

Sending an Email with GmailApp

```
function sendEmail() {  
  GmailApp.sendEmail('recipient@example.com', 'Test Email', 'Hello, this is a test email  
from Google Apps Script.');
```

Adding an Event to Google Calendar

```
function addCalendarEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  calendar.createEvent('New Event', new Date(), new Date());  
}
```

Fetching Data from an External API

```
function fetchDataFromAPI() {  
  var response = UrlFetchApp.fetch('https://api.example.com/data');  
  Logger.log(response.getContentText());  
}
```

Creating a Custom Menu in Google Sheets

```
function onOpen() {  
  var ui = SpreadsheetApp.getUi();  
  ui.createMenu('Custom Menu')  
  .addItem('Say Hello', 'sayHello')  
  .addToUi();
```

```
}  
function sayHello() {  
  
  SpreadsheetApp.getActiveSpreadsheet().getActiveSheet().getRange('B1').setValue('Hello!');  
}
```

Using Google Sheets as a Database

```
function logData() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var rowContents = ['John', 'Doe', new Date()];  
  sheet.appendRow(rowContents);  
}
```

Updating Google Sheet Data Based on Conditions

```
function updateCells() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getRange('A1:A10');  
  var values = range.getValues();  
  for (var i = 0; i < values.length; i++) {  
    if (values[i][0] === 'Done') {  
      sheet.getRange(i + 1, 2).setValue('Completed');  
    }  
  }  
}
```

Formatting Cells in Google Sheets

```
function formatCells() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getRange('B1:B5');  
  range.setBackground('yellow');  
  range.setFontWeight('bold');  
}
```

Removing Duplicate Rows in Google Sheets

```
function removeDuplicates() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var data = sheet.getDataRange().getValues();  
  var newData = [];  
  var unique = {};  
  for (var i = 0; i < data.length; i++) {  
    var row = data[i];  
    var key = row.join("-"); // Create a unique string key for comparison
```

```

if (!unique[key]) {
  unique[key] = true;
  newData.push(row);
}
}
sheet.clearContents(); // Clear the old data
sheet.getRange(1, 1, newData.length, newData[0].length).setValues(newData); // Set
new data
}

```

Inserting a Chart in Google Sheets

```

function insertChart() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getRange('A1:B10'); // Select data range for the chart
  var chart = sheet.newChart()
  .setChartType(Charts.ChartType.LINE)
  .addRange(range)
  .setPosition(5, 5, 0, 0)
  .build();
  sheet.insertChart(chart);
}

```

Convert Spreadsheet Data to JSON

```

function convertSheetToJson() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var values = range.getValues();
  var keys = values.shift(); // First row as keys
  var json = [];
  for (var i = 0; i < values.length; i++) {
    var row = {};
    for (var j = 0; j < keys.length; j++) {
      row[keys[j]] = values[i][j];
    }
    json.push(row);
  }
  Logger.log(JSON.stringify(json));
}

```

Batch Update Google Sheet Cells

```

function batchUpdateCells() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var rangeList = sheet.getRangeList(['A1', 'B2', 'C3']); // List of cell addresses
  rangeList.setValue('Updated');
}

```

```
}
```

Auto-Resize Columns in Google Sheets

```
function autoResizeColumns() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  sheet.autoResizeColumns(1, sheet.getLastColumn()); // Auto-resize all columns  
}
```

Find and Replace Text in Google Docs

```
function findAndReplaceText() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  body.replaceText('oldText', 'newText');  
}
```

List All Files in Google Drive Folder

```
function listFilesInFolder() {  
  var folderId = 'your-folder-id-here'; // Replace with your folder ID  
  var files = DriveApp.getFolderById(folderId).getFiles();  
  while (files.hasNext()) {  
    var file = files.next();  
    Logger.log(file.getName());  
  }  
}
```

Create a New Google Drive Folder

```
function createNewFolder() {  
  var newFolder = DriveApp.createFolder('New Folder');  
  Logger.log('Folder created with ID: ' + newFolder.getId());  
}
```

Send an Email with an Attachment from Google Drive

```
function sendEmailWithAttachment() {  
  var file = DriveApp.getFileById('your-file-id-here'); // Replace with your file ID  
  GmailApp.sendEmail(  
    'recipient@example.com',  
    'Email with Attachment',  
    'Please find the attached file.',  
    {  
      attachments: [file.getAs(MimeType.PDF)], // Change MimeType as needed  
      name: 'Automatic Emailer Script'  
    }  
  );  
}
```



```
);  
}
```

Add Custom Footer to Google Doc

```
function addCustomFooter() {  
  var doc = DocumentApp.getActiveDocument();  
  var footer = doc.addFooter(); // Add a footer if it doesn't exist  
  var paragraph = footer.appendParagraph('Confidential Document - For Internal Use  
Only');  
  paragraph.setAlignment(DocumentApp.HorizontalAlignment.CENTER);  
}
```

Archiving Emails from Gmail

```
function archiveReadEmails() {  
  var threads = GmailApp.getInboxThreads();  
  for (var i = 0; i < threads.length; i++) {  
    if (threads[i].isUnread() === false) {  
      threads[i].moveToArchive();  
    }  
  }  
}
```

Create a Document Copy with a New Name

```
function createDocumentCopy() {  
  var originalDocId = 'original-doc-id'; // Replace with your original document ID  
  var originalDoc = DriveApp.getFileById(originalDocId);  
  var copy = originalDoc.makeCopy('New Document Copy');  
  Logger.log('Copied document ID: ' + copy.getId());  
}
```

Inserting a New Row in Google Sheets After the Last Row with Data

```
function insertNewRow() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var lastRow = sheet.getLastRow();  
  sheet.insertRowAfter(lastRow);  
  sheet.getRange(lastRow + 1, 1).setValue('New Data');  
}
```

Retrieve and Log the Names of All Spreadsheets in Google Drive

```
function logAllSpreadsheets() {
  var files = DriveApp.getFilesByType(MimeType.GOOGLE_SHEETS);
  while (files.hasNext()) {
    var file = files.next();
    Logger.log(file.getName());
  }
}
```

Add a Custom Status Column to Google Sheets Based on Another Column's Values

```
function addStatusColumn() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getRange("A2:A"); // Assuming column A contains data to check
  var values = range.getValues();
  for (var i = 0; i < values.length; i++) {
    if (values[i][0] === 'Completed') {
      sheet.getRange(i + 2, 2).setValue('Done');
    } else {
      sheet.getRange(i + 2, 2).setValue('In Progress');
    }
  }
}
```

Automatically Color Code Rows in Google Sheets Based on Cell Value

```
function colorCodeRows() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange(); // Select all used cells
  var values = range.getValues();
  for (var i = 0; i < values.length; i++) {
    var row = sheet.getRange(i + 1, 1, 1, sheet.getLastColumn());
    if (values[i][0] === 'High Priority') {
      row.setBackground('red');
    } else if (values[i][0] === 'Medium Priority') {
      row.setBackground('yellow');
    } else if (values[i][0] === 'Low Priority') {
      row.setBackground('green');
    }
  }
}
```

Create and Email a PDF from a Google Sheet

```
function emailSheetAsPDF() {
```

```

var sheet = SpreadsheetApp.getActiveSpreadsheet();
var sheetId = sheet.getActiveSheet().getSheetId();
var url = 'https://docs.google.com/spreadsheets/d/' + sheet.getId() +
'/export?format=pdf&gid=' + sheetId;
var options = {
headers: {
'Authorization': 'Bearer ' + ScriptApp.getOAuthToken()
}
};
var response = UrlFetchApp.fetch(url, options);
var blob = response.getBlob().setName(sheet.getName() + '.pdf');
GmailApp.sendEmail('recipient@example.com', 'Sheet PDF', 'Find the attached PDF.', {
attachments: [blob]
});
}

```

Update Google Calendar Events Based on Google Sheets Data

```

function updateCalendarEvents() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var dataRange = sheet.getDataRange();
var data = dataRange.getValues();
var calendar = CalendarApp.getCalendarById('your-calendar-id@example.com'); //
Replace with your Calendar ID
for (var i = 1; i < data.length; i++) { // Starting from row 2 to skip headers
var eventId = data[i][0]; // Assuming the event ID is in the first column
var newTitle = data[i][1]; // Assuming the new title is in the second column
var event = calendar.getEventById(eventId);
if (event) {
event.setTitle(newTitle);
}
}
}
}

```

Export Google Sheets Data to a New Google Doc

```

function exportDataToDoc() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var values = sheet.getDataRange().getValues();
var doc = DocumentApp.create('Exported Data Document');
var body = doc.getBody();
values.forEach(function(row) {
var line = row.join(' '); // Tab-separated values
body.appendParagraph(line);
});
}

```

```
}
```

Automatically Backup Google Sheets to Another Spreadsheet

```
function backupSheet() {  
  var sourceSheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var sourceData = sourceSheet.getDataRange().getValues();  
  var backupSpreadsheet = SpreadsheetApp.openById('your-backup-spreadsheet-id'); //  
  Replace with your backup spreadsheet ID  
  var backupSheet = backupSpreadsheet.getSheetByName('Backup') ||  
  backupSpreadsheet.insertSheet('Backup');  
  backupSheet.clear(); // Clear existing data  
  backupSheet.getRange(1, 1, sourceData.length,  
  sourceData[0].length).setValues(sourceData);  
}
```

Setting Validation Rules in Google Sheets

```
function setValidationRules() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getRange("B1:B10"); // Specify the range for validation  
  // Create the data validation rule  
  var rule = SpreadsheetApp.newDataValidation()  
  .requireValueInList(['Yes', 'No'], true)  
  .setAllowInvalid(false)  
  .setHelpText('Select Yes or No')  
  .build();  
  range.setDataValidation(rule);  
}
```

Creating a Google Calendar Event with Guests

```
function createCalendarEventWithGuests() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var startTime = new Date();  
  var endTime = new Date(startTime.getTime() + 3600000); // Adds 1 hour  
  var options = {  
    guests: 'guest1@example.com,guest2@example.com',  
    sendInvites: true  
  };  
  calendar.createEvent('Meeting with Guests', startTime, endTime, options);  
}
```

Appending a Row to a Google Sheet from Form Submission

```
function appendRowFromForm() {
  var formResponse = ['John Doe', 'johndoe@example.com', 'Feedback']; // Simulate
  form response
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  sheet.appendRow(formResponse);
}
```

Automatically Formatting New Rows in Google Sheets

```
function formatNewRows() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var lastRow = sheet.getLastRow();
  var range = sheet.getRange(lastRow, 1, 1, 3); // Adjust column length as needed
  // Set the background color and text alignment for the new row
  range.setBackground('#ffffcc').setHorizontalAlignment('center');
}
```

Fetching and Logging Weather Data

```
function fetchWeatherData() {
  var response =
  UrlFetchApp.fetch('http://api.openweathermap.org/data/2.5/weather?q=London&appid=
  your_api_key'); // Replace 'your_api_key' with your actual API key
  var weatherData = JSON.parse(response.getContentText());
  Logger.log(weatherData.main.temp); // Log the temperature
}
```

Bulk Removing Rows Based on Condition in Google Sheets

```
function bulkRemoveRows() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues();
  for (var i = data.length - 1; i >= 0; i--) {
    if (data[i][0] === 'Remove') { // Check first column for 'Remove'
      sheet.deleteRow(i + 1);
    }
  }
}
```

Sending Customized Email Based on Google Sheets Data

```
function sendCustomEmails() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var data = range.getValues();
  for (var i = 1; i < data.length; i++) { // Skip header row
```

```

var emailAddress = data[i][1]; // Assuming email addresses are in the second column
var message = 'Dear ' + data[i][0] + ', your custom message here.'; // Assuming names
are in the first column
var subject = 'Your Custom Subject';
GmailApp.sendEmail(emailAddress, subject, message);
}
}

```

Sync Google Sheets Data with Google Calendar

```

function syncSheetWithCalendar() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var eventsRange = sheet.getRange("A2:C10"); // Adjust range to your event details
var events = eventsRange.getValues();
var calendar = CalendarApp.getDefaultCalendar();
events.forEach(function(event) {
var title = event[0];
var startTime = new Date(event[1]);
var endTime = new Date(event[2]);
calendar.createEvent(title, startTime, endTime);
});
}

```

Generating a Report from Google Sheets and Emailing as a PDF

```

function generateAndEmailReport() {
var sheet = SpreadsheetApp.getActiveSpreadsheet();
var sheetId = sheet.getId();
var sheetName = sheet.getName();
var pdfUrl = "https://docs.google.com/spreadsheets/d/" + sheetId +
"/export?exportFormat=pdf&format=pdf";
var options = {
headers: {
'Authorization': 'Bearer ' + ScriptApp.getOAuthToken()
}
};
var response = UrlFetchApp.fetch(pdfUrl, options);
var blob = response.getBlob().setName(sheetName + '.pdf');
GmailApp.sendEmail("recipient@example.com", "Monthly Report", "Please find the
attached report.", {
attachments: [blob],
});
}

```

Removing Duplicate Entries in Google Sheets

```

function removeDuplicates() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues();
  var unique = {};
  var uniqueRows = [];
  for (var i = 0; i < data.length; i++) {
    var row = data[i];
    var key = row.join("|"); // Create a unique key from the row values
    if (!unique[key]) {
      unique[key] = true;
      uniqueRows.push(row);
    }
  }
  sheet.clearContents(); // Clear the old data
  sheet.getRange(1, 1, uniqueRows.length,
  uniqueRows[0].length).setValues(uniqueRows); // Set the unique data
}

```

Splitting a String into Multiple Cells in Google Sheets

```

function splitStringIntoCells() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getRange("A1"); // Assuming the string to be split is in A1
  var stringValue = range.getValue();
  var splitValues = stringValue.split(","); // Split by comma, change delimiter as needed
  // Set the split values starting from B1
  for (var i = 0; i < splitValues.length; i++) {
    sheet.getRange(1, i + 2).setValue(splitValues[i]);
  }
}

```

Merging Multiple Google Sheets into One Sheet

```

function mergeSheets() {
  var destinationSpreadsheet = SpreadsheetApp.getActiveSpreadsheet();
  var destinationSheet = destinationSpreadsheet.getSheetByName('Merged') ||
  destinationSpreadsheet.insertSheet('Merged');
  var spreadsheets = ['spreadsheetId1', 'spreadsheetId2']; // Add spreadsheet IDs
  for (var i = 0; i < spreadsheets.length; i++) {
    var sheet = SpreadsheetApp.openById(spreadsheets[i]).getActiveSheet();
    var data = sheet.getDataRange().getValues();
    destinationSheet.getRange(destinationSheet.getLastRow() + 1, 1, data.length,
    data[0].length).setValues(data);
  }
}

```

Highlighting Cells Based on Expiry Date

```
function highlightExpiryDates() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange(); // Select the range where expiry dates are stored
  var values = range.getValues();
  for (var i = 0; i < values.length; i++) {
    var expiryDate = new Date(values[i][1]); // Assuming expiry dates are in the second
    column
    var today = new Date();
    if (expiryDate < today) {
      sheet.getRange(i + 1, 2).setBackground('red'); // Highlight expired items
    }
  }
}
```

Automatically Add a Timestamp to a Row on Edit

```
function onEdit(e) {
  var range = e.range;
  var sheet = range.getSheet();
  if (sheet.getName() === 'Sheet1' && range.getColumn() == 1 && e.value) { // Check if
  edit is in column A of Sheet1
  var timestampCell = sheet.getRange(range.getRow(), 2); // Set timestamp in column B
  timestampCell.setValue(new Date());
  }
}
```

Convert Google Sheet Rows to JSON Objects

```
function convertRowsToJson() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var values = range.getValues();
  var headers = values.shift(); // Remove the first row with headers
  var jsonObjects = [];
  for (var i = 0; i < values.length; i++) {
    var jsonObject = {};
    for (var j = 0; j < headers.length; j++) {
      jsonObject[headers[j]] = values[i][j];
    }
    jsonObjects.push(jsonObject);
  }
  Logger.log(JSON.stringify(jsonObjects));
}
```


Creating a Function to Clear Specific Cells in Google Sheets

```
function clearSpecificCells() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var cellsToClear = ['B2', 'C4', 'D6']; // Specify the cells to clear  
  cellsToClear.forEach(function(cellAddress) {  
    sheet.getRange(cellAddress).clearContent();  
  });  
}
```

Function to Count Non-Empty Cells in a Specific Column

```
function countNonEmptyCells() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var column = sheet.getRange('A:A'); // Specify the column to count  
  var values = column.getValues();  
  var count = 0;  
  values.forEach(function(row) {  
    if (row[0]) count++;  
  });  
  Logger.log('Non-empty cells: ' + count);  
}
```

Sending a Daily Digest Email from a Google Sheet

```
function sendDailyDigest() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var dataRange = sheet.getDataRange();  
  var data = dataRange.getValues();  
  var message = "";  
  data.forEach(function(row, index) {  
    if (index > 0) { // Skip header row  
      message += row.join(', ') + '\n'; // Format each row's values separated by commas  
    }  
  });  
  GmailApp.sendEmail('recipient@example.com', 'Daily Digest', message);  
}
```

Auto-Resizing Row Height in Google Sheets Based on Content

```
function autoResizeRowHeight() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getDataRange();  
  var numRows = range.getNumRows();
```

```
for (var i = 1; i <= numRows; i++) {  
  sheet.autoResizeRow(i);  
}  
}
```

Function to Replace Text in a Google Doc

```
function replaceTextInDoc() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  body.replaceText('oldText', 'newText'); // Replace 'oldText' with 'newText'  
}
```

Batch Update Cell Background Color in Google Sheets

```
function batchUpdateCellBackground() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getRange("A1:C10"); // Specify your range  
  var backgrounds = [];  
  for (var i = 0; i < 10; i++) { // Assume 10 rows for example  
    backgrounds.push(["#FF0000", "#00FF00", "#0000FF"]); // Red, Green, Blue  
  }  
  range.setBackgrounds(backgrounds);  
}
```

Create a List of Hyperlinks in Google Sheets

```
function createListOfHyperlinks() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var urls = [  
    ['Google', 'https://www.google.com'],  
    ['Bing', 'https://www.bing.com'],  
    ['DuckDuckGo', 'https://www.duckduckgo.com']  
  ]; // Array of [text, hyperlink] pairs  
  var range = sheet.getRange(1, 1, urls.length, 2); // Adjust range according to the size of  
  urls  
  var formulas = urls.map(function(url) {  
    return ["=HYPERLINK("" + url[1] + "", "" + url[0] + "");"];  
  });  
  range.setFormulas(formulas);  
}
```

Summarize Data from Multiple Sheets into a Master Sheet

```
function summarizeData() {  
  var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
```

```

var summarySheet = spreadsheet.getSheetByName("Summary") ||
spreadsheet.insertSheet("Summary");
var sheets = spreadsheet.getSheets();
var summaryData = [];
sheets.forEach(function(sheet) {
if (sheet.getName() !== "Summary") {
var data = sheet.getRange("A1:B10").getValues(); // Adjust your range
summaryData = summaryData.concat(data);
}
});
summarySheet.getRange(1, 1, summaryData.length, 2).setValues(summaryData);
}

```

Convert Column Index to Letter in Google Sheets

```

function columnToLetter(column) {
var temp, letter = "";
while (column > 0) {
temp = (column - 1) % 26;
letter = String.fromCharCode(temp + 65) + letter;
column = (column - temp - 1) / 26;
}
Logger.log(letter);
}

```

Automate Email Sending Based on Cell Value Changes

```

function onEditTrigger(e) {
var range = e.range;
var sheet = range.getSheet();
if (sheet.getName() === "Emails" && range.getColumn() == 2 &&
e.value.toLowerCase() === 'yes') { // Check if edit is in the right column and sheet
var emailAddress = sheet.getRange(range.getRow(), 1).getValue(); // Assuming email
addresses are in the first column
GmailApp.sendEmail(emailAddress, 'Notification', 'Your custom message here.');
```

Extract Email Addresses from Text

```

function extractEmails() {
var text = 'Your long text with emails@example.com somewhere in between';
var regex = /[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}/g;
var emails = text.match(regex);
Logger.log(emails);
}

```

Generate a Random Password Generator in Google Sheets

```
function generateRandomPassword() {  
  var length = 12; // Define the length of the password  
  var charset =  
  "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";  
  var password = "";  
  for (var i = 0; i < length; i++) {  
    var randomIndex = Math.floor(Math.random() * charset.length);  
    password += charset.charAt(randomIndex);  
  }  
  Logger.log(password);  
}
```

Delete Rows Based on a Condition in Google Sheets

```
function deleteRowsBasedOnCondition() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var lastRow = sheet.getLastRow();  
  for (var i = lastRow; i >= 1; i--) {  
    var cellValue = sheet.getRange(i, 1).getValue(); // Check values in the first column  
    if (cellValue === 'Delete') { // Change 'Delete' to your condition  
      sheet.deleteRow(i);  
    }  
  }  
}
```

Create a Table of Contents for a Google Doc

```
function createTableOfContents() {  
  var doc = DocumentApp.getActiveDocument();  
  var body = doc.getBody();  
  var headings = body.getParagraphs().filter(function(paragraph) {  
    return paragraph.getHeading() !== DocumentApp.ParagraphHeading.NORMAL;  
  });  
  var toc = doc.addTable();  
  headings.forEach(function(heading, index) {  
    toc.appendTableRow([heading.getText(), 'Page ' + (index + 1)]);  
  });  
}
```

Updating Google Calendar Event Descriptions in Bulk

```
function updateCalendarEventDescriptions() {  
  var calendar = CalendarApp.getDefaultCalendar();
```

```

var events = calendar.getEvents(new Date("2023-01-01"), new Date("2023-12-31")); //
Specify your date range
for (var i = 0; i < events.length; i++) {
  var event = events[i];
  event.setDescription('Updated Description: ' + event.getDescription());
}
}

```

Highlighting Weekends in Google Sheets

```

function highlightWeekends() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var numColumns = range.getNumColumns();
  var numRows = range.getNumRows();
  var dates = range.getValues();
  for (var i = 0; i < numRows; i++) {
    for (var j = 0; j < numColumns; j++) {
      var cellDate = dates[i][j];
      if (cellDate instanceof Date) {
        var dayOfWeek = cellDate.getDay();
        if (dayOfWeek == 0 || dayOfWeek == 6) { // 0 = Sunday, 6 = Saturday
          sheet.getRange(i + 1, j + 1).setBackground('#FFF2CC');
        }
      }
    }
  }
}

```

Inserting 'Last Updated' Timestamp in Google Sheets Footer

```

function insertLastUpdatedTimestamp() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var footer = "Last Updated: " + new Date();
  sheet.getRange('A1').setValue(footer); // Change 'A1' to the cell where you want the
timestamp
}

```

Copy Google Sheet Data to Another Spreadsheet

```

function copyDataToAnotherSpreadsheet() {
  var sourceSheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var targetSpreadsheetId = 'your-target-spreadsheet-id';
  var targetSheetName = 'Sheet1';
  var sourceRange = sourceSheet.getDataRange();
  var sourceValues = sourceRange.getValues();
}

```

```

var targetSpreadsheet = SpreadsheetApp.openById(targetSpreadsheetId);
var targetSheet = targetSpreadsheet.getSheetByName(targetSheetName);
targetSheet.getRange(1, 1, sourceValues.length,
sourceValues[0].length).setValues(sourceValues);
}

```

Clear Contents from Multiple Ranges in Google Sheets

```

function clearMultipleRanges() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var rangesToClear = ['B2:D4', 'G5:H10', 'A12:A15']; // Define the ranges to clear
rangesToClear.forEach(function(range) {
sheet.getRange(range).clearContent();
});
}

```

Create Multiple Google Calendar Events from Google Sheets Data

```

function createMultipleCalendarEvents() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var dataRange = sheet.getDataRange();
var eventDetails = dataRange.getValues();
var calendar = CalendarApp.getDefaultCalendar();
for (var i = 1; i < eventDetails.length; i++) { // Assuming the first row is headers
var [title, startDate, endDate] = eventDetails[i];
calendar.createEvent(title, new Date(startDate), new Date(endDate));
}
}

```

Validate Email Format in Google Sheets Using Regex

```

function validateEmailFormat() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var range = sheet.getRange("A1:A10"); // Adjust the range accordingly
var values = range.getValues();
var regexPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/; // Simple email validation pattern
for (var i = 0; i < values.length; i++) {
if (!regexPattern.test(values[i][0])) {
sheet.getRange(i + 1, 2).setValue("Invalid"); // Mark as invalid in the next column
} else {
sheet.getRange(i + 1, 2).setValue("Valid");
}
}
}
}

```

Automatically Resize Images in Google Docs

```
function resizeImagesInDoc() {
  var doc = DocumentApp.getActiveDocument();
  var body = doc.getBody();
  var images = body.getImages();
  images.forEach(function(image) {
    var width = image.getWidth();
    var height = image.getHeight();
    var newWidth = 200; // Set the desired width
    var scaleFactor = newWidth / width;
    image.setWidth(newWidth);
    image.setHeight(height * scaleFactor);
  });
}
```

Convert Google Sheets Range to HTML Table

```
function convertRangeToHtmlTable() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var values = range.getValues();
  var htmlTable = "<table><tr>";
  // Add headers
  values[0].forEach(function(header) {
    htmlTable += "<th>" + header + "</th>";
  });
  htmlTable += "</tr>";
  // Add row data
  for (var i = 1; i < values.length; i++) {
    htmlTable += "<tr>";
    values[i].forEach(function(cell) {
      htmlTable += "<td>" + cell + "</td>";
    });
    htmlTable += "</tr>";
  }
  htmlTable += "</table>";
  Logger.log(htmlTable);
}
```

Automatically Sort Data in Google Sheets When a Cell Changes

```
function onEdit(e) {
  var range = e.range;
  var sheet = range.getSheet();
```

```

if (sheet.getName() === 'SortSheet' && range.getColumn() == 1) { // Change
'SortSheet' to your sheet name
var dataRange = sheet.getRange("A2:B10"); // Change to the range you want to sort
dataRange.sort({column: 1, ascending: true}); // Sort by the first column in ascending
order
}
}

```

Fetch Latest Cryptocurrency Prices and Update Google Sheets

```

function updateCryptoPrices() {
var apiURL =
'https://api.coingecko.com/api/v3/simple/price?ids=bitcoin,ethereum&vs_currencies=usd
';
var response = UrlFetchApp.fetch(apiURL);
var json = JSON.parse(response.getContentText());
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
sheet.getRange('A1').setValue('Bitcoin');
sheet.getRange('A2').setValue('Ethereum');
sheet.getRange('B1').setValue(json.bitcoin.usd);
sheet.getRange('B2').setValue(json.ethereum.usd);
}

```

Compare Two Columns and Highlight Mismatches in Google Sheets

```

function highlightMismatches() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var range1 = sheet.getRange("A1:A10"); // First column range
var range2 = sheet.getRange("B1:B10"); // Second column range
var values1 = range1.getValues();
var values2 = range2.getValues();
for (var i = 0; i < values1.length; i++) {
if (values1[i][0] !== values2[i][0]) {
range1.getCell(i + 1, 1).setBackground('yellow');
range2.getCell(i + 1, 1).setBackground('yellow');
} else {
range1.getCell(i + 1, 1).setBackground(null);
range2.getCell(i + 1, 1).setBackground(null);
}
}
}
}

```


Automatically Hide Rows with Zero Values in Google Sheets

```
function hideZeroRows() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var dataRange = sheet.getDataRange();
  var values = dataRange.getValues();
  for (var i = values.length - 1; i >= 0; i--) {
    if (values[i].some(function(value) { return value === 0; })) {
      sheet.hideRows(i + 1);
    } else {
      sheet.showRows(i + 1);
    }
  }
}
```

Send a Scheduled Email Reminder from Google Sheets Data

```
function sendScheduledReminder() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var dataRange = sheet.getRange("A2:B10"); // Assuming A has dates and B has email addresses
  var today = new Date();
  var data = dataRange.getValues();
  data.forEach(function(row) {
    var date = new Date(row[0]);
    var email = row[1];
    if (date.toDateString() === today.toDateString()) {
      GmailApp.sendEmail(email, 'Reminder', 'This is your scheduled reminder.');
```

Export Specific Google Sheets Data to a New Sheet

```
function exportDataToNewSheet() {
  var sourceSheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var targetSpreadsheet = SpreadsheetApp.create('New Spreadsheet for Export');
  var targetSheet = targetSpreadsheet.getActiveSheet();
  var dataRange = sourceSheet.getRange("A1:C10"); // Adjust range as needed
  var data = dataRange.getValues();
  targetSheet.getRange(1, 1, data.length, data[0].length).setValues(data);
}
```

Validate Data Entry as Email Format in Google Sheets

```
function validateEmailEntry() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getRange("B1:B10"); // Adjust range for your needs
  var rule = SpreadsheetApp.newDataValidation().requireTextIsEmail().build();
  range.setDataValidation(rule);
}
```

Log Changes in Google Sheets to a Separate Sheet

```
function logChanges(e) {
  var range = e.range;
  var sheet = range.getSheet();
  var logSheet = sheet.getParent().getSheetByName('Change Log') ||
  sheet.getParent().insertSheet('Change Log');
  var oldValue = e.oldValue || 'None';
  var newValue = e.value || 'None';
  var timestamp = new Date();
  var user = Session.getActiveUser().getEmail();
  logSheet.appendRow([timestamp, user, range.getA1Notation(), oldValue, newValue]);
}
```

Generate QR Codes in Google Sheets

```
function generateQRcodes() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var dataRange = sheet.getRange("A2:A"); // Adjust the range according to where your
  data is
  var values = dataRange.getValues();
  for (var i = 0; i < values.length; i++) {
    var data = values[i][0];
    if (data) {
      var url = "https://chart.googleapis.com/chart?cht=qr&chl=" +
      encodeURIComponent(data) + "&chs=100x100";
      sheet.getRange(i + 2, 2).setValue(url); // Assuming you want the QR code links in the
      second column
    }
  }
}
```

Summarize Google Forms Responses in Google Sheets

```
function summarizeFormResponses() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Form
  Responses 1'); // Change to your form response sheet name
  var dataRange = sheet.getDataRange();
  var responses = dataRange.getValues();
  var summary = {};
```

```

// Assuming first row is headers, skip it
for (var i = 1; i < responses.length; i++) {
  var response = responses[i];
  for (var j = 0; j < response.length; j++) {
    var question = responses[0][j];
    var answer = response[j];
    summary[question] = summary[question] || {};
    summary[question][answer] = (summary[question][answer] || 0) + 1;
  }
}
// Log summary for review or further processing
Logger.log(summary);
}

```

Auto-Generate Document Based on Google Sheets Data

```

function generateDocumentFromSheetData() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getRange("A1:B1").getValues(); // Adjust range as needed
  var doc = DocumentApp.create('Generated Document from Sheet');
  var body = doc.getBody();
  data.forEach(function(row) {
    body.appendParagraph(row[0] + ': ' + row[1]); // Assuming key-value pairs in each row
  });
  Logger.log('Document created with ID: ' + doc.getId());
}

```

Create and Format a New Google Sheet from Script

```

function createAndFormatSheet() {
  var newSheetName = 'Monthly Report'; // Change as required
  var spreadsheet = SpreadsheetApp.create(newSheetName);
  var sheet = spreadsheet.getActiveSheet();
  // Set up header row
  var headers = ['Date', 'Category', 'Amount'];
  var headerRange = sheet.getRange("A1:C1");
  headerRange.setValues([headers]);
  headerRange.setFontWeight('bold');
  sheet.setColumnWidths(1, 3, 150); // Set widths for the first three columns
  Logger.log('Spreadsheet created with name: ' + newSheetName);
}

```

Change Google Sheets Tab Color Based on Cell Value

```

function updateTabColorBasedOnValue() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();

```

```

var status = sheet.getRange('A1').getValue(); // Assuming status is in A1
if (status === 'Complete') {
  sheet.setTabColor('green');
} else if (status === 'In Progress') {
  sheet.setTabColor('yellow');
} else {
  sheet.setTabColor(null); // Remove color if status is neither
}
}

```

Automatically Number Rows in Google Sheets

```

function autoNumberRows() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange(); // Get range of data
  var values = range.getValues();
  for (var i = 0; i < values.length; i++) {
    values[i][0] = i + 1; // Assuming the row number should go in the first column
  }
  range.setValues(values);
}

```

Remove Empty Rows from the End of Google Sheets

```

function removeEmptyRows() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var lastRow = sheet.getLastRow();
  var lastColumn = sheet.getLastColumn();
  var range = sheet.getRange(lastRow, 1, 1, lastColumn);
  var values = range.getValues();
  while (values[0].every(function(value) { return value === ""; })) {
    sheet.deleteRow(lastRow);
    lastRow--;
    range = sheet.getRange(lastRow, 1, 1, lastColumn);
    values = range.getValues();
  }
}

```

Convert Selected Range to Uppercase in Google Sheets

```

function convertRangeToUppercase() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getActiveRange(); // Get the current selected range
  var values = range.getValues();
  var newValues = values.map(function(row) {
    return row.map(function(cell) {
      return typeof cell === 'string' ? cell.toUpperCase() : cell;
    });
  });
}

```

```
});  
});  
range.setValues(newValues);  
}
```

Sync Google Sheets Data with a Secondary Sheet

```
function syncSheets() {  
  var primarySheet =  
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Primary");  
  var secondarySheet =  
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Secondary");  
  var data = primarySheet.getDataRange().getValues();  
  secondarySheet.clear(); // Clear existing data  
  secondarySheet.getRange(1, 1, data.length, data[0].length).setValues(data); // Copy  
  data to secondary sheet  
}
```

Generate a List of All Hyperlinks in Google Sheets

```
function listAllHyperlinks() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getDataRange();  
  var formulas = range.getFormulas();  
  var links = [];  
  for (var i = 0; i < formulas.length; i++) {  
    for (var j = 0; j < formulas[i].length; j++) {  
      var cellFormula = formulas[i][j];  
      if (cellFormula.startsWith('=HYPERLINK')) {  
        links.push(cellFormula);  
      }  
    }  
  }  
  Logger.log(links);  
}
```

Create Google Calendar Events from Selected Range in Google Sheets

```
function createEventsFromRange() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getActiveRange(); // Get the current selected range  
  var values = range.getValues();  
  var calendar = CalendarApp.getDefaultCalendar();  
  values.forEach(function(row) {  
    var title = row[0];  
    var startTime = new Date(row[1]);
```

```

var endTime = new Date(row[2]);
calendar.createEvent(title, startTime, endTime);
});
}

```

Highlight Duplicate Entries in Google Sheets

```

function highlightDuplicates() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var values = range.getValues();
  var duplicates = [];
  values.forEach(function(row, rowIndex) {
    row.forEach(function(cell, columnIndex) {
      if (values.slice(rowIndex + 1).some(r => r[columnIndex] === cell)) {
        duplicates.push([rowIndex + 1, columnIndex + 1]);
      }
    });
  });
  duplicates.forEach(function(cellPosition) {
    sheet.getRange(cellPosition[0], cellPosition[1]).setBackground('red');
  });
}

```

Automatically Format New Google Sheets Entries as Currency

```

function formatAsCurrency() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getRange("B1:B"); // Assuming column B should be in currency
  format
  range.setNumberFormat("$#,##0.00");
}

```

Compare Two Google Sheets for Differences

```

function compareSheets() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet1 = ss.getSheetByName("Sheet1");
  var sheet2 = ss.getSheetByName("Sheet2");
  var range1 = sheet1.getDataRange();
  var range2 = sheet2.getDataRange();
  var values1 = range1.getValues();
  var values2 = range2.getValues();
  var maxRows = Math.max(values1.length, values2.length);
  var maxCols = Math.max(values1[0].length, values2[0].length);
  for (var i = 0; i < maxRows; i++) {

```



```
var dateCell = sheet.getRange(editRange.getRow(), editRange.getLastColumn() + 1); //  
Assuming date is always next to the edited cell  
dateCell.setValue(new Date());  
}
```

Automatically Convert Incoming Emails to Google Docs

```
function convertEmailsToDocs() {  
  var threads = GmailApp.getInboxThreads(0, 5); // Fetch up to 5 latest email threads  
  threads.forEach(function(thread) {  
    var messages = thread.getMessages();  
    messages.forEach(function(message) {  
      var doc = DocumentApp.create('Email from: ' + message.getFrom());  
      doc.getBody().setText(message.getPlainBody());  
    });  
  });  
}
```

Create a Pie Chart in Google Sheets from Data

```
function createPieChart() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getRange("A1:B6"); // Adjust range for your data  
  var chart = sheet.newChart()  
  .setChartType(Charts.ChartType.PIE)  
  .addRange(range)  
  .setPosition(5, 5, 0, 0)  
  .build();  
  sheet.insertChart(chart);  
}
```

Send a Custom Slack Message from a Google Sheet

```
function sendSlackMessage() {  
  var webhookUrl = 'your-slack-webhook-url'; // Replace with your Slack webhook URL  
  var message = {  
    "text": "Here is a message from Google Sheets!"  
  };  
  var options = {  
    "method": "post",  
    "contentType": "application/json",  
    "payload": JSON.stringify(message)  
  };  
  UrlFetchApp.fetch(webhookUrl, options);  
}
```

Update a Cell Based on Changes in Another Cell


```

function onEdit(e) {
  var sourceSheet = e.source.getActiveSheet();
  if (sourceSheet.getName() === 'Sheet1') { // Specify the sheet to watch
    var editedCell = sourceSheet.getActiveRange();
    if (editedCell.getA1Notation() === 'A1') { // Check if the edited cell is A1
      var newValue = editedCell.getValue();
      var targetCell = sourceSheet.getRange('B1'); // Specify the target cell
      targetCell.setValue(newValue * 2); // Example action, multiply A1 value by 2
    }
  }
}

```

Batch Create Google Calendar Events from Spreadsheet Data

```

function batchCreateEvents() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var startRow = 2; // Assuming the first row has headers
  var numRows = sheet.getLastRow() - 1;
  var dataRange = sheet.getRange(startRow, 1, numRows, 4); // Adjust 4 to the number
  of columns
  var data = dataRange.getValues();
  var calendar = CalendarApp.getDefaultCalendar();
  for (var i = 0; i < data.length; i++) {
    var row = data[i];
    var title = row[0]; // First column
    var startTime = new Date(row[1]); // Second column
    var endTime = new Date(row[2]); // Third column
    var options = {description: row[3]}; // Fourth column for description
    calendar.createEvent(title, startTime, endTime, options);
  }
}

```

Color Code Cells Based on Expiry Date in Google Sheets

```

function colorCodeExpiryDates() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getRange("A2:B10"); // Assuming A column has dates and B column
  is to be colored
  var values = range.getValues();
  var today = new Date();
  for (var i = 0; i < values.length; i++) {
    var expiryDate = new Date(values[i][0]);
    var daysLeft = (expiryDate - today) / (1000 * 3600 * 24);
    if (daysLeft < 0) {
      sheet.getRange('B' + (i + 2)).setBackground('red'); // Past expiry date
    }
  }
}

```

```

} else if (daysLeft < 30) {
sheet.getRange('B' + (i + 2)).setBackground('yellow'); // Expiring soon
} else {
sheet.getRange('B' + (i + 2)).setBackground('green'); // Valid
}
}
}
}

```

Auto-Delete Rows Based on Cell Content in Google Sheets

```

function autoDeleteRows() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var range = sheet.getDataRange();
var values = range.getValues();
for (var i = values.length - 1; i >= 0; i--) {
if (values[i][0] === "Delete") { // Assuming the trigger word is in the first column
sheet.deleteRow(i + 1);
}
}
}
}

```

Convert All Formulas to Values in Google Sheets

```

function convertFormulasToValues() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var range = sheet.getDataRange(); // Select all data in the sheet
var values = range.getValues(); // Get all values, formulas are evaluated
range.setValues(values); // Set evaluated values back, converting all formulas to static values
}
}

```

Auto-Generate and Email a Google Sheets Report as PDF

```

function emailSheetAsPDF() {
var ss = SpreadsheetApp.getActiveSpreadsheet();
var sheet = ssgetActiveSheet();
var sheetId = sheet.getSheetId();
var url = 'https://docs.google.com/spreadsheets/d/' + ss.getId() +
'/export?format=pdf&gid=' + sheetId;
var response = UrlFetchApp.fetch(url, {
headers: { 'Authorization': 'Bearer ' + ScriptApp.getOAuthToken() }
});
var blob = response.getBlob().setName(sheet.getName() + '.pdf');
GmailApp.sendEmail('recipient@example.com', 'Your Google Sheets Report', 'Please
find the attached report.', {
attachments: [blob]
}
}

```

```
});  
}
```

Set Data Validation for a Range Based on Another Column's Values

```
function setDataValidationBasedOnColumn() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getRange('B1:B10'); // Range to apply validation to  
  var criteriaRange = sheet.getRange('A1:A10'); // Range to take validation criteria from  
  var rule = SpreadsheetApp.newDataValidation().requireValueInRange(criteriaRange,  
  true).build();  
  range.setDataValidation(rule);  
}
```

Highlight Cells in Google Sheets That Contain Formulas

```
function highlightFormulas() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var range = sheet.getDataRange();  
  var formulas = range.getFormulas();  
  for (var i = 0; i < formulas.length; i++) {  
    for (var j = 0; j < formulas[i].length; j++) {  
      if (formulas[i][j]) { // If cell contains a formula  
        sheet.getRange(i + 1, j + 1).setBackground('#FFFF00'); // Highlight with yellow  
      }  
    }  
  }  
}
```

Automatically Add Google Calendar Events to a Google Sheet

```
function importCalendarEvents() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEvents(new Date(), new Date(new Date().getTime() + 7 * 24 *  
  3600 * 1000)); // Fetch events for the next 7 days  
  events.forEach(function(event) {  
    sheet.appendRow([event.getTitle(), event.getStartTime(), event.getEndTime()]);  
  });  
}
```

Create a Google Sheets Log for Failed Email Attempts

```
function logFailedEmails() {
```

```

var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Email Log');
var data = [['john.doe@example.com', 'Failed'], ['jane.doe@example.com', 'Success']];
// Sample data
data.forEach(function(row) {
  if (row[1] === 'Failed') {
    sheet.appendRow(row); // Log only failed email attempts
  }
});
}

```

Batch Clear Multiple Named Ranges in Google Sheets

```

function clearNamedRanges() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var namedRanges = ss.getNamedRanges();
  namedRanges.forEach(function(namedRange) {
    namedRange.getRange().clearContent(); // Clear content of each named range
  });
}

```

Merge Multiple Google Docs into One Document

```

function mergeGoogleDocs() {
  var docIds = ['doc-id-1', 'doc-id-2', 'doc-id-3']; // Array of Google Docs IDs to merge
  var finalDoc = DocumentApp.create('Merged Document');
  docIds.forEach(function(id) {
    var body = DocumentApp.openById(id).getBody();
    var elements = body.getParagraphs();
    elements.forEach(function(element) {
      finalDoc.getBody().appendParagraph(element.copy());
    });
    finalDoc.getBody().appendPageBreak();
  });
}

```

Auto-Format Negative Numbers in Red in Google Sheets

```

function formatNegativeNumbers() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var values = range.getValues();
  values.forEach(function(row, rowIndex) {
    row.forEach(function(cell, columnIndex) {
      if (typeof cell === 'number' && cell < 0) {
        sheet.getRange(rowIndex + 1, columnIndex + 1).setFontColor('red');
      }
    });
  });
}

```

```
});  
}
```

Extract and Summarize Checked Items from Google Forms Checkbox Questions

```
function summarizeCheckedItems() {  
  var form = FormApp.getActiveForm();  
  var formResponses = form.getResponses();  
  var itemResponses = formResponses.map(response =>  
  response.getItemResponses());  
  var summary = {};  
  itemResponses.forEach(response => {  
    response.forEach(itemResponse => {  
      var answers = itemResponse.getResponse();  
      if (Array.isArray(answers)) { // Check if response is from a checkbox question  
        answers.forEach(answer => {  
          summary[answer] = (summary[answer] || 0) + 1;  
        });  
      }  
    });  
  });  
  Logger.log(summary); // Log the summary of checked items  
}
```

Automate Data Backups in Google Sheets

```
function backupSheetData() {  
  const sourceSheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  const backupSpreadsheet = SpreadsheetApp.openById('your-backup-spreadsheet-id');  
  // Replace with your backup spreadsheet ID  
  const backupSheet = backupSpreadsheet.getSheetByName('Backup') ||  
  backupSpreadsheet.insertSheet('Backup');  
  const data = sourceSheet.getDataRange().getValues();  
  backupSheet.clear(); // Clear the existing data  
  backupSheet.getRange(1, 1, data.length, data[0].length).setValues(data); // Copy new  
  data  
}
```

Track Cell Edits and Timestamps in a Separate Google Sheets Tab

```
function onEdit(e) {  
  const editSheet = e.sourcegetActiveSheet();  
  const logSheet = e.source.getSheetByName('Edit Log') || e.source.insertSheet('Edit  
  Log');
```

```

const timestamp = new Date();
const editRange = e.range.getA1Notation();
const oldValue = e.oldValue || 'Empty';
const newValue = e.value || 'Empty';
logSheet.appendRow([timestamp, editSheet.getName(), editRange, oldValue,
newValue]);
}

```

Summarize Selected Google Sheets Data in an Email

```

function summarizeAndEmailData() {
const sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
const range = sheet.getDataRange();
const values = range.getValues();
let summary = 'Your Data Summary:\n\n';
// Customize this loop for your specific summarization needs
values.forEach((row, index) => {
summary += `Row ${index + 1}: ${row.join(', ')}\n`;
});
GmailApp.sendEmail('recipient@example.com', 'Google Sheets Data Summary',
summary);
}

```

Find and Replace Text in Multiple Google Docs

```

function findAndReplaceInDocs() {
const docIds = ['doc-id-1', 'doc-id-2']; // Array of document IDs
const findText = 'oldText';
const replaceText = 'newText';
docIds.forEach((id) => {
const body = DocumentApp.openById(id).getBody();
body.replaceText(findText, replaceText);
});
}

```

Create a Dropdown List from Range in Google Sheets

```

function createDropdownFromRange() {
const sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
const range = sheet.getRange('A1:A10'); // Range for the dropdown list
const validationRule =
SpreadsheetApp.newDataValidation().requireValueInRange(range, true).build();
sheet.getRange('B1').setDataValidation(validationRule); // Cell to have dropdown
}

```

Automatically Move Google Sheets Rows to Another Tab Based on Cell Value

```
function moveRowsBasedOnValue() {
  const sourceSheet =
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Source');
  const targetSheet =
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Target');
  const rows = sourceSheet.getDataRange().getValues();
  rows.forEach((row, index) => {
    if (row[0] === 'Move') { // Check if the first cell in the row has 'Move'
      targetSheet.appendRow(row); // Append row to target sheet
      sourceSheet.deleteRow(index + 1); // Remove row from source sheet
    }
  });
}
```

Generate and Insert a Google Sheets Chart into a Google Slide

```
function insertChartIntoSlides() {
  const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Data');
  const slides = SlidesApp.openById('your-slides-id'); // Replace with your Google Slides ID
  const slide = slides.getSlides()[0]; // Get the first slide
  const dataRange = sheet.getRange('A1:B5'); // Adjust range to your data
  const chart = sheet.newChart()
  .setChartType(Charts.ChartType.BAR)
  .addRange(dataRange)
  .setPosition(5, 5, 0, 0)
  .build();
  sheet.insertChart(chart);
  const embeddedChart = chart.getBlob();
  slide.insertImage(embeddedChart);
}
```

Extract All Email Addresses from a Gmail Thread

```
function extractEmailsFromThread() {
  const thread = GmailApp.getInboxThreads(0, 1)[0]; // Get the first thread in your inbox
  const messages = thread.getMessages();
  const emailAddresses = [];
  messages.forEach((message) => {
    const from = message.getFrom();
    const to = message.getTo();
    const cc = message.getCc();
  });
}
```

```

emailAddresses.push(from, to, cc);
});
Logger.log(emailAddresses.join(', ')); // Log all email addresses found
}

```

Auto-Create a Monthly Calendar in Google Sheets

```

function createMonthlyCalendar() {
  const sheet = SpreadsheetApp.getActiveSpreadsheet().insertSheet('Monthly
  Calendar');
  const days = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];
  const today = new Date();
  const firstDay = new Date(today.getFullYear(), today.getMonth(), 1);
  const lastDay = new Date(today.getFullYear(), today.getMonth() + 1, 0);
  const dates = [];
  // Set up days header
  sheet.getRange('A1:G1').setValues([days]).setFontWeight('bold');
  // Generate dates for the month
  for (let d = firstDay; d <= lastDay; d.setDate(d.getDate() + 1)) {
    dates.push(new Date(d));
  }
  // Populate the calendar
  let row = 2;
  for (let i = 0; i < dates.length; i++) {
    const column = dates[i].getDay() + 1;
    sheet.getRange(row, column).setValue(dates[i].getDate());
    if (column === 7) row++;
  }
}

```

Monitor and Log Google Sheets Cell Changes Over Time

```

function logCellChanges(e) {
  const changeLogSheet =
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Change Log') ||
  SpreadsheetApp.getActiveSpreadsheet().insertSheet('Change Log');
  const cell = e.range.getA1Notation();
  const oldValue = e.oldValue || 'None';
  const newValue = e.value || 'None';
  const timestamp = new Date();
  // Log the change
  changeLogSheet.appendRow([timestamp, cell, oldValue, newValue]);
}

```

Automatically Color-code Google Calendar Events Based on Keywords


```

function colorCodeCalendarEvents() {
  var calendar = CalendarApp.getDefaultCalendar();
  var events = calendar.getEvents(new Date(), new Date(new Date().getTime() + 30 * 24
* 3600 * 1000)); // Next 30 days
  events.forEach(function(event) {
    var title = event.getTitle().toLowerCase();
    if (title.includes('birthday')) {
      event.setColor(CalendarApp.EventColor.PALE_BLUE);
    } else if (title.includes('meeting')) {
      event.setColor(CalendarApp.EventColor.YELLOW);
    } else if (title.includes('deadline')) {
      event.setColor(CalendarApp.EventColor.RED);
    }
  });
}

```

Create a Directory of Files and Folders in Google Drive

```

function createDriveDirectoryListing() {
  var folderId = 'your-folder-id'; // Replace with your folder ID
  var folder = DriveApp.getFolderById(folderId);
  var contents = folder.getFiles();
  var listing = [['Name', 'Type', 'URL', 'Last Updated']];
  while (contents.hasNext()) {
    var file = contents.next();
    listing.push([file.getName(), file.getMimeType(), file.getUrl(), file.getLastUpdated()]);
  }
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  sheet.getRange(1, 1, listing.length, 4).setValues(listing);
}

```

Export Google Sheets Charts as Images

```

function exportChartsAsImages() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var charts = sheet.getCharts();
  charts.forEach(function(chart, index) {
    var blob = chart.getBlob();
    DriveApp.createFile(blob.setName('Chart_' + (index + 1) + '.png'));
  });
}

```

Sync Google Sheets Data with Google Contacts

```

function syncSheetWithContacts() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues(); // Assuming first row is headers

```

```

data.forEach(function(row, index) {
  if (index > 0) { // Skip header row
    var contact = ContactsApp.createContact(row[0], row[1], row[2]); // Assuming Name,
    Last Name, Email in columns
    contact.addPhone(ContactsApp.Field.MOBILE_PHONE, row[3]); // Assuming phone
    number in fourth column
  }
});
}

```

Auto-Sort Google Sheets When a New Row is Added

```

function onEdit(event) {
  var sheet = event.source.getActiveSheet();
  var editedCell = sheet.getActiveCell();
  if (editedCell.getRow() == 1 && sheet.getName() == 'YourSheetName') { // Check if edit
  is in the first row and correct sheet
    var range = sheet.getRange('A2:Z' + sheet.getLastRow());
    range.sort({column: 1, ascending: true}); // Sort by first column, A-Z
  }
}

```

Remove Duplicate Rows Based on a Specific Column in Google Sheets

```

function removeDuplicatesBasedOnColumn() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues();
  var uniqueRecords = {};
  var rowsToDelete = [];
  data.forEach(function(row, index) {
    if (uniqueRecords.hasOwnProperty(row[0])) { // Assuming duplicates based on the first
    column
      rowsToDelete.push(index + 1);
    } else {
      uniqueRecords[row[0]] = true;
    }
  });
  rowsToDelete.reverse(); // Delete from the bottom to avoid messing up indices
  rowsToDelete.forEach(function(rowIndex) {
    sheet.deleteRow(rowIndex);
  });
}

```

Insert Current Weather Information into Google Sheets

```

function insertCurrentWeather() {

```

```

var apiKey = 'your-openweather-api-key'; // Replace with your OpenWeather API key
var city = 'London,uk'; // Replace with your city
var url = 'http://api.openweathermap.org/data/2.5/weather?q=' + city + '&appid=' +
apiKey + '&units=metric';
var response = UrlFetchApp.fetch(url);
var json = JSON.parse(response.getContentText());
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
sheet.getRange('A1').setValue('Current Temperature (°C)');
sheet.getRange('A2').setValue(json.main.temp);
}

```

Convert Google Sheet Rows to Individual Text Files

```

function convertRowsToTextFiles() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var rows = sheet.getDataRange().getValues();
rows.forEach(function(row, index) {
var text = row.join(', '); // Combine all cell values in the row, separated by commas
var fileName = 'Row_' + (index + 1) + '.txt';
DriveApp.createFile(fileName, text);
});
}

```

Monitor Google Sheets Cell Value and Send Alert if Changed

```

function monitorCellValue() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var cellToMonitor = sheet.getRange('A1').getValue(); // Change 'A1' to the cell you want
to monitor
var triggerValue = 100; // Change 100 to the value that triggers the alert
if (cellToMonitor > triggerValue) {
MailApp.sendEmail('your-email@example.com', 'Alert: Cell Value Changed', 'The value
of cell A1 exceeds ' + triggerValue);
}
}

```

Create a Checklist in Google Sheets with Checkboxes

```

function createChecklist() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var tasks = ['Task 1', 'Task 2', 'Task 3']; // Replace with your tasks
var range = sheet.getRange('A1:A' + tasks.length);
range.setValues(tasks.map(function(task) { return [task]; }));
range.offset(0, 1).insertCheckboxes();
}

```

Auto-generate Google Calendar Events from Task List in Google Sheets

```
function generateCalendarEventsFromTasks() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues(); // Assuming tasks start from row 2
  var calendar = CalendarApp.getDefaultCalendar();
  data.forEach(function(row, index) {
    if (index > 0 && row[0] && !row[4]) { // Assuming tasks in column A and completion flag
      in column E
      var title = row[0];
      var startDate = new Date(row[1]);
      var endDate = new Date(row[2]);
      var description = row[3];
      var event = calendar.createEvent(title, startDate, endDate, {description: description});
      sheet.getRange(index + 1, 5).setValue('Created'); // Mark as 'Created'
    }
  });
}
```

Highlight Overdue Tasks in Google Sheets

```
function highlightOverdueTasks() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var tasks = sheet.getRange('A2:B' + sheet.getLastRow()).getValues(); // Assuming due
  dates are in column B
  tasks.forEach(function(row, i) {
    var dueDate = row[1];
    if (dueDate instanceof Date && dueDate < new Date()) {
      sheet.getRange(i + 2, 2).setBackground('red'); // Highlight overdue tasks
    }
  });
}
```

Convert Sheet Data to JSON and Log It

```
function sheetDataToJson() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues();
  var headers = data.shift(); // Remove the first row with headers
  var jsonData = [];
  data.forEach(function(row) {
    var obj = {};
    headers.forEach(function(header, i) {
      obj[header] = row[i];
    });
    jsonData.push(obj);
  });
}
```

```
});  
Logger.log(JSON.stringify(jsonData));  
}
```

Batch Resize Images in Google Drive

```
function batchResizeImages() {  
  var folder = DriveApp.getFolderById('your-folder-id'); // Replace with your folder ID  
  var images = folder.GetFilesByType(MimeType.JPEG); // Change MIME type as needed  
  while (images.hasNext()) {  
    var image = images.next();  
    var blob = image.getBlob();  
    var resizedImage = ImagesService.newImage(blob).resize(100, 100).getBlob(); //  
    Resize to 100x100 pixels  
    folder.createFile(resizedImage).setName('Resized_' + image.getName());  
  }  
}
```

Insert Formatted Date and Time in a Cell

```
function insertFormattedDateTime() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var cell = sheet.getRange('A1'); // Change to the target cell  
  var now = new Date();  
  var formattedDate = Utilities.formatDate(now, Session.getScriptTimeZone(),  
  'yyyy-MM-dd HH:mm:ss');  
  cell.setValue(formattedDate);  
}
```

Validate Email Addresses in a Column

```
function validateEmailAddresses() {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  var emails = sheet.getRange('A2:A' + sheet.getLastRow()).getValues(); // Assuming  
  emails in column A  
  var emailRegex = /^[^s@]+@[^s@]+\.[^s@]+$/;  
  emails.forEach(function(email, i) {  
    if (!emailRegex.test(email[0])) {  
      sheet.getRange(i + 2, 1).setBackground('yellow'); // Highlight invalid emails  
    }  
  });  
}
```

Automatically Close Google Form Submissions After a Certain Number of Responses

```
function closeFormAfterResponses() {
```

```

var form = FormApp.getActiveForm();
var maxResponses = 100; // Set the maximum number of responses
var responses = form.getResponses().length;
if (responses >= maxResponses) {
  form.setAcceptingResponses(false).setCustomClosedFormMessage('We have reached
the maximum number of responses. Thank you!');
}
}

```

Create a Progress Tracker in Google Sheets

```

function createProgressTracker() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var tasks = ['Task 1', 'Task 2', 'Task 3', 'Task 4']; // Example tasks
  var totalTasks = tasks.length;
  sheet.getRange('A1').setValue('Task');
  sheet.getRange('B1').setValue('Status');
  tasks.forEach(function(task, index) {
    sheet.getRange('A' + (index + 2)).setValue(task);
    sheet.getRange('B' + (index + 2)).setValue('=IF(C' + (index + 2) + '<='', '', IF(C' + (index
+ 2) + '< 100, "In Progress", "Completed"))');
    sheet.getRange('C' + (index +
2)).setValue('').setDataValidation(SpreadsheetApp.newDataValidation().requireNumber
Between(0, 100).build()).setBackground('#dddddd');
  });
  sheet.setColumnWidth(3, 100);
  sheet.getRange('C1').setValue('Progress (%)');
  sheet.getRange('A1:C' + (totalTasks +
1)).applyRowBanding(SpreadsheetApp.BandingTheme.LIGHT_GREY);
}

```

Generate Unique ID for Each Google Sheets Row

```

function generateUniqueIds() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getRange('A2:A' + sheet.getLastRow()); // Assuming IDs to be in
column A
  var values = range.getValues();
  values.forEach(function(row, index) {
    if (!row[0]) { // If ID cell is empty
      var uniqueId = Utilities.getUuid(); // Generate a unique ID
      sheet.getRange('A' + (index + 2)).setValue(uniqueId);
    }
  });
}

```

Extract Specific Data from Gmail Messages to Google Sheets

```
function extractDataFromGmail() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var threads = GmailApp.search('subject:"Your Subject Filter"'); // Adjust search query as
  needed
  var messages = GmailApp.getMessagesForThreads(threads);
  messages.forEach(function(thread, i) {
    thread.forEach(function(message) {
      var subject = message.getSubject();
      var date = message.getDate();
      var body = message.getPlainBody();
      // Extract specific information from the email body using regex, if necessary
      sheet.appendRow([subject, date, body]); // Adjust according to the data you want to
      extract
    });
  });
}
```

Automatically Update Google Sheets with Stock Prices

```
function updateStockPrices() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var symbols = ['AAPL', 'GOOGL', 'MSFT']; // Example stock symbols
  var prices = [];
  symbols.forEach(function(symbol) {
    var url = 'https://api.iextrading.com/1.0/stock/' + symbol + '/price';
    var response = UrlFetchApp.fetch(url);
    var price = parseFloat(response.getContentText());
    prices.push([symbol, price]);
  });
  sheet.getRange(1, 1, prices.length, 2).setValues(prices);
}
```

Automatically Archive Gmail Messages in Google Sheets

```
function archiveGmailMessages() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var threads = GmailApp.search('label:inbox is:unread'); // Example search query
  var messages = GmailApp.getMessagesForThreads(threads);
  messages.forEach(function(thread) {
    thread.forEach(function(message) {
      var subject = message.getSubject();
      var sender = message.getFrom();
      var date = message.getDate();
      sheet.appendRow([subject, sender, date]);
    });
  });
}
```

```
message.markRead();
});
});
}
```

Automatically Generate Google Calendar Events from Google Forms Responses

```
function createEventsFromFormResponses() {
  var form = FormApp.openById('your-form-id'); // Replace with your Form ID
  var responses = form.getResponses();
  var calendar = CalendarApp.getDefaultCalendar();
  responses.forEach(function(response) {
    var itemResponses = response.getItemResponses();
    var title = itemResponses[0].getResponse(); // Assuming first question is event title
    var startTime = new Date(itemResponses[1].getResponse()); // Assuming second
question is start time
    var endTime = new Date(itemResponses[2].getResponse()); // Assuming third question
is end time
    calendar.createEvent(title, startTime, endTime);
  });
}
```

Automatically Send Email Notifications for Google Forms Responses

```
function sendEmailNotificationsForFormResponses() {
  var form = FormApp.openById('your-form-id'); // Replace with your Form ID
  var responses = form.getResponses();
  responses.forEach(function(response) {
    var respondentEmail = response.getRespondentEmail();
    var itemResponses = response.getItemResponses();
    var message = 'Thank you for submitting the form. Your response has been received.';
    var subject = 'Form Submission Confirmation';
    MailApp.sendEmail(respondentEmail, subject, message);
  });
}
```

Automatically Create Google Calendar Events from Google Sheets Data

```
function createCalendarEventsFromSheetData() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues();
  var calendar = CalendarApp.getDefaultCalendar();
  data.forEach(function(row) {
```



```

var title = row[0]; // Assuming event title in first column
var startTime = new Date(row[1]); // Assuming start time in second column
var endTime = new Date(row[2]); // Assuming end time in third column
calendar.createEvent(title, startTime, endTime);
});
}

```

Automatically Track Changes to Google Sheets with Revision History

```

function trackSheetChanges() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var revisions = sheet.getRevisions();
revisions.forEach(function(revision) {
Logger.log('Revision ID: ' + revision.getRevisionId());
Logger.log('User: ' + revision.getUser());
Logger.log('Timestamp: ' + revision.getTimestamp());
Logger.log('Changes: ' + revision.getChanges());
});
}

```

Automatically Generate QR Codes from URLs in Google Sheets

```

function generateQRCodesFromURLs() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var urls = sheet.getRange('A:A').getValues(); // Assuming URLs are in column A
urls.forEach(function(url, index) {
var qrCode = 'https://api.qrserver.com/v1/create-qr-code/?data=' +
encodeURIComponent(url);
sheet.insertImage(qrCode, index + 1, 2); // Insert QR code in column B
});
}

```

Automatically Generate Google Docs from Google Sheets Data

```

function generateDocsFromSheetData() {
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var data = sheet.getDataRange().getValues();
data.forEach(function(row) {
var title = row[0]; // Assuming document title in first column
var body = row[1]; // Assuming document content in second column
DocumentApp.create(title).getBody().setText(body);
});
}

```

Automatically Translate Google Sheets Data

```
function translateSheetData() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var range = sheet.getDataRange();
  var values = range.getValues();
  var translatedValues = [];
  values.forEach(function(row) {
    var translatedRow = [];
    row.forEach(function(cell) {
      var translatedCell = LanguageApp.translate(cell, 'en', 'fr'); // Translate from English to
      French
      translatedRow.push(translatedCell);
    });
    translatedValues.push(translatedRow);
  });
  range.setValues(translatedValues);
}
```

Automatically Generate Google Slides Presentations from Google Sheets Data

```
function generateSlidesFromSheetData() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = sheet.getDataRange().getValues();
  var slides = SlidesApp.create('Presentation from Sheet Data');
  var presentation = slides.getSlides();
  data.forEach(function(row) {
    var slide = slides.appendSlide();
    var slideTitle = row[0]; // Assuming slide title in first column
    var slideBody = row[1]; // Assuming slide content in second column
    slide.insertText(0, 0, slideBody);
  });
}
```