

# Apps Script Questions with Solutions and Coding Examples 3



How can I copy a sheet from one Google Sheet to another using Google Apps Script?	2
How can I protect a range of cells in a Google Sheet using Google Apps Script?	2
How can I get the current date and time in a Google Sheet using Google Apps Script?	3
How can I sort a range of cells in a Google Sheet using Google Apps Script?	3
How can I get the values of all cells in a range in a Google Sheet using Google Apps Script?	4
How can I format a range of cells in a Google Sheet using Google Apps Script?	4
How can I get the email addresses of all members in a Google Group using Google Apps Script?	5
How can I create a new calendar event in Google Calendar using Google Apps Script?	6
How can I send an email using Gmail in Google Apps Script?	7
How can I create a new Google Document using Google Apps Script?	7

## How can I copy a sheet from one Google Sheet to another using Google Apps Script?

```
function copySheet() {  
  var ss = SpreadsheetApp.getActiveSpreadsheet();  
  var sourceSheet = ss.getSheetByName('Source Sheet');  
  var targetSpreadsheet =  
SpreadsheetApp.openById('Target Spreadsheet ID');  
  sourceSheet.copyTo(targetSpreadsheet);  
}
```

Explanation: This code copies a sheet named 'Source Sheet' from the active Google Sheet to a target Google Sheet with the specified ID ('Target Spreadsheet ID') using the copyTo() function of the sheet object.

## How can I protect a range of cells in a Google Sheet using Google Apps Script?

```
function protectRange() {  
  var sheet = SpreadsheetApp.getActiveSheet();  
  var range = sheet.getRange('A1:B5');  
  var protection =  
range.protect().setDescription('Protected Range');  
  protection.addEditor('user@example.com');  
  protection.removeEditors(protection.getEditors());  
  if (protection.canDomainEdit()) {  
    protection.setDomainEdit(false);  
  }  
}
```

Explanation: This code protects a range of cells (A1:B5) in the active sheet of a Google Sheet using the `protect()` function of the range object. The `setDescription()` function is used to set a description for the protection. The `addEditor()` function is used to add a user (user@example.com) to the list of editors who can edit the protected range. The `removeEditors()` function is used to remove all editors except the specified user. The `canDomainEdit()` function is used to check if the protection allows editing by users in the domain. If so, the `setDomainEdit()` function is used to disable domain editing.

## How can I get the current date and time in a Google Sheet using Google Apps Script?

```
function getCurrentDateTime() {  
    var date = new Date();  
    var sheet = SpreadsheetApp.getActiveSheet();  
    sheet.getRange('A1').setValue(date);  
}
```

Explanation: This code gets the current date and time using the `Date()` function, and then sets the value of cell A1 in the active sheet of a Google Sheet to the current date and time using the `setValue()` function of the range object.

## How can I sort a range of cells in a Google Sheet using Google Apps Script?

```
function sortRange() {
```

```
var sheet = SpreadsheetApp.getActiveSheet();  
var range = sheet.getRange('A2:C6');  
range.sort(1); // Sort by column 1 (ascending)  
}
```

Explanation: This code sorts a range of cells (A2:C6) in the active sheet of a Google Sheet in ascending order based on the values in column 1 (the first column) using the `sort()` function of the range object.

## How can I get the values of all cells in a range in a Google Sheet using Google Apps Script?

```
function getRangeValues() {  
  var sheet = SpreadsheetApp.getActiveSheet();  
  var range = sheet.getRange('A1:C3');  
  var values = range.getValues();  
  Logger.log(values);  
}
```

Explanation: This code gets the values of all cells in a range (A1:C3) in the active sheet of a Google Sheet using the `getValues()` function of the range object. The `Logger.log()` function is used to log the values to the execution log.

## How can I format a range of cells in a Google Sheet using Google Apps Script?

```
function formatRange() {  
  var sheet = SpreadsheetApp.getActiveSheet();
```

```

var range = sheet.getRange('A1:C3');
range.setBackground('red');
range.setFontColor('white');
range.setFontWeight('bold');
range.setHorizontalAlignment('center');
range.setVerticalAlignment('middle');
}

```

Explanation: This code formats a range of cells (A1:C3) in the active sheet of a Google Sheet using various formatting functions of the range object. The `setBackground()` function is used to set the background color of the range to red. The `setFontColor()` function is used to set the font color of the range to white. The `setFontWeight()` function is used to set the font weight of the range to bold. The `setHorizontalAlignment()` function is used to set the horizontal alignment of the range to center. The `setVerticalAlignment()` function is used to set the vertical alignment of the range to middle.

## How can I get the email addresses of all members in a Google Group using Google Apps Script?

```

function getGroupMembers() {
  var group =
GroupsApp.getGroupByEmail('example@googlegroups.com');
  var members = group.getUsers();
  for (var i = 0; i < members.length; i++) {
    Logger.log(members[i].getEmail());
  }
}

```

Explanation: This code gets the email addresses of all members in a Google Group (example@googlegroups.com) using the `getGroupByEmail()` function of the Groups service. The `getUsers()` function is used to get an array of user objects, and then a for loop is used to iterate over the array and log the email addresses of the users to the execution log using the `getEmail()` function of the user object.

## How can I create a new calendar event in Google Calendar using Google Apps Script?

```
function createEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var event = calendar.createEvent('New Event',  
    new Date('2023-04-12T10:00:00Z'),  
    new Date('2023-04-12T11:00:00Z'),  
    {description: 'Event Description', location: 'Event  
Location'});  
  Logger.log('Event ID: ' + event.getId());  
}
```

Explanation: This code creates a new calendar event in the default Google Calendar using the `createEvent()` function of the Calendar service. The `createEvent()` function takes four arguments: the event title ('New Event'), the start time of the event (`new Date('2023-04-12T10:00:00Z')`), the end time of the event (`new Date('2023-04-12T11:00:00Z')`), and an object containing additional parameters for the event (`{description: 'Event Description', location: 'Event Location'}`). The `getId()` function is used to get the ID of the created event, which is then logged to the execution log.

## How can I send an email using Gmail in Google Apps Script?

```
function sendEmail() {  
  var recipient = 'recipient@example.com';  
  var subject = 'Email Subject';  
  var body = 'Email Body';  
  GmailApp.sendEmail(recipient, subject, body);  
}
```

Explanation: This code sends an email to a recipient ('recipient@example.com') using the sendEmail() function of the Gmail service. The sendEmail() function takes three arguments: the email recipient, the email subject, and the email body.

## How can I create a new Google Document using Google Apps Script?

```
function createDocument() {  
  var document = DocumentApp.create('New Document');  
  var body = document.getBody();  
  var paragraph = body.appendParagraph('Hello,  
world!');  
  
  paragraph.setHeading(DocumentApp.ParagraphHeading.HEADING1);  
  Logger.log('Document URL: ' + document.getUrl());  
}
```

Explanation: This code creates a new Google Document using the `create()` function of the Document service. The `create()` function takes a single argument, which is the title of the new document ('New Document'). The `getBody()` function is used to get the body of the new document, and the `appendParagraph()` function is used to add a new paragraph to the body with the text 'Hello, world!'. The `setHeading()` function is used to set the heading of the paragraph to `HEADING1`. Finally, the `getUrl()` function is used to get the URL of the new document, which is then logged to the execution log.