Apps Script Questions with Solutions and Coding Examples 6

How can I format the font size of a range in a Google Sheet using Google Apps Script? 1
How can I format the boldness of a range in a Google Sheet using Google Apps Script? 2
How can I format the alignment of text in a range in a Google Sheet using Google Apps Script? 3
How can I freeze rows or columns in a Google Sheet using Google Apps Script? 3
How can I sort a range in a Google Sheet using Google Apps Script? 4
How can I filter a range in a Google Sheet using Google Apps Script? 5
How can I copy data from one sheet to another in a Google Sheet using Google Apps Script? 6
How can I send an email from a Google Sheet using Google Apps Script? 7
How can I send an email with an attachment from a Google Sheet using Google Apps Script? 7
How can I create a new Google Doc from a Google Sheet using Google Apps Script? 8

How can I add a new row to a Google Sheet using Google Apps Script? 9

How can I format the font size of a range in a Google Sheet using Google Apps Script?

```
function setRangeFontSize() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1:D4');
  range.setFontSize(14); // set to 14 point font
}
```

Explanation: This code formats the font size of a range in a Google Sheet using the setFontSize() function of the Range object. The getSheetByName() function is used to get a reference to the sheet containing the data. The getRange() function is used to get a range object representing the range of data to be formatted. Finally, the setFontSize() function is used to set the font size of the range to 14 points.

How can I format the boldness of a range in a Google Sheet using Google Apps Script?

```
function setRangeBoldness() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1:D4');
  range.setFontWeight('bold'); // set to bold
}
```

Explanation: This code formats the boldness of a range in a Google Sheet using the setFontWeight() function of the Range object. The getSheetByName() function is used to get a reference to the sheet containing the data. The getRange() function is used to get a range object representing the range of data to be formatted. Finally, the setFontWeight() function is used to set the boldness of the range to bold.

How can I format the alignment of text in a range in a Google Sheet using Google Apps Script?

```
function setRangeTextAlignment() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1:D4');
  range.setHorizontalAlignment('center'); // center
align
  range.setVerticalAlignment('middle'); // middle align
}
```

Explanation: This code formats the alignment of text in a range in a Google Sheet using the setHorizontalAlignment() and setVerticalAlignment() functions of the Range object. The getSheetByName() function is used to get a reference to the sheet containing the data. The getRange() function is used to get a range object representing the range of data to be formatted. Finally, the setHorizontalAlignment() function is used to set the horizontal alignment of the range to center, and the setVerticalAlignment() function is used to set the vertical alignment of the range to middle.

How can I freeze rows or columns in a Google Sheet using Google Apps Script?

```
function freezeRowsAndColumns() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  sheet.setFrozenRows(1); // freeze the first row
  sheet.setFrozenColumns(2); // freeze the first two
columns
}
```

Explanation: This code freezes rows or columns in a Google Sheet using the setFrozenRows() and setFrozenColumns() functions of the Sheet object. The getSheetByName() function is used to get a reference to the sheet containing the data. The setFrozenRows() function is used to freeze the first row of the sheet, and the setFrozenColumns() function is used to freeze the first two columns of the sheet.

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

```
function sortRange() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Data Sheet');
  var range = sheet.getRange('A2:C10');
  range.sort(1); // sort by the first column in ascending order
}
```

Explanation: This code sorts a range in a Google Sheet using the sort() function of the Range object. The getSheetByName() function is used to get a reference to the sheet containing the data. The getRange() function is used to get a range object representing the range of data to be sorted. Finally, the sort() function is used to sort the range by the first column in ascending order.

How can I filter a range in a Google Sheet using Google Apps Script?

```
function filterRange() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Data Sheet');
  var range = sheet.getRange('A1:C10');
  var criteria = SpreadsheetApp.newFilterCriteria().whenTextStartsWith('A').build();
  var filter = range.createFilter().setColumnFilterCriteria(1, criteria); // filter by the first column starting with "A"
}
```

Explanation: This code filters a range in a Google Sheet using the createFilter() and setColumnFilterCriteria() functions of the Range object. The getSheetByName() function is used to get a reference to the sheet containing the data. The getRange() function is used to get a range object representing the range of data to be filtered. The newFilterCriteria() function is used to create a filter criteria object that specifies the text that the first column should start with. Finally, the createFilter() function is used to create a filter based on the filter criteria object, and the setColumnFilterCriteria() function is used to set the filter for the first column of the range.

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

To copy data from one sheet to another in a Google Sheet using Google Apps Script, you can use the copyTo() function of the Range object. Here is an example code:

```
function copyData() {
  var sourceSheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Source Sheet');
  var sourceRange = sourceSheet.getRange('A1:B10');
  var targetSheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Target Sheet');
  var targetRange = targetSheet.getRange('C1');
  sourceRange.copyTo(targetRange);
}
```

Explanation:

1. The getSheetByName() function is used to get a reference to the source sheet containing the data that you want to copy.
2. The getRange() function is used to get a reference to the range of cells that you want to copy from the source sheet.
3. The getSheetByName() function is used to get a reference to the target sheet where you want to paste the data.
4. The getRange() function is used to get a reference to the cell in the target sheet where you want to paste the data.
5. The copyTo() function is used to copy the data from the source range to the target range.

Note: The copyTo() function can also be used with other options like formatOnly, contentsOnly, and destinationSpreadsheetId.

How can I send an email from a Google Sheet using Google Apps Script?

```
function sendEmail() {
  var recipient = 'recipient@example.com';
  var subject = 'Subject of the email';
  var body = 'Body of the email';
  MailApp.sendEmail(recipient, subject, body);
}
```

Explanation: This code sends an email from a Google Sheet using the sendEmail() function of the MailApp object. The recipient, subject, and body variables are used to specify the recipient email address, subject, and body of the email, respectively. The sendEmail() function is then used to send the email.

How can I send an email with an attachment from a Google Sheet using Google Apps Script?

```
function sendEmailWithAttachment() {
  var recipient = 'recipient@example.com';
  var subject = 'Subject of the email';
  var body = 'Body of the email';
  var file = DriveApp.getFileById('fileId'); // replace
fileId with the ID of the file to be attached
  MailApp.sendEmail({
    to: recipient,
    subject: subject,
    body: body,
    attachments: [file.getAs(MimeType.PDF)]
  });
}
```

Explanation: This code sends an email with an attachment from a Google Sheet using the sendEmail() function of the MailApp object. The recipient, subject, and body variables are used to specify the recipient email address, subject, and body of the email, respectively. The DriveApp.getFileById() function is used to get a reference to the file to be attached, and the getAs() function is used to specify the MIME type of the file. Finally, the sendEmail() function is used to send the email with the attachment.

How can I create a new Google Doc from a Google Sheet using Google Apps Script?

```
function createNewDoc() {
  var docName = 'Name of the new Google Doc';
  var doc = DocumentApp.create(docName);
  var body = doc.getBody();
  body.appendParagraph('This is the first paragraph of the new Google Doc.');
}
```

Explanation: This code creates a new Google Doc from a Google Sheet using the create() function of the DocumentApp object. The docName variable is used to specify the name of the new Google Doc. The create() function is then used to create the new Google Doc, and the getBody() function is used to get a reference to the body of the document. Finally, the appendParagraph() function is used to add a new paragraph to the body of the document.

How can I add a new row to a Google Sheet using Google Apps Script?

```
function addNewRow() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Data Sheet');
  var values = ['Value 1', 'Value 2', 'Value 3'];
  sheet.appendRow(values);
}
```

Explanation: This code adds a new row to a Google Sheet using the appendRow() function of the Sheet object. The getSheetByName() function is used to get a reference to the sheet where the new row should be added. The values variable is used to specify the values to be added to the new row. Finally, the appendRow() function is used to add the new row to the sheet.