Apps Script Questions with Solutions and Coding Examples 7

How can I send an email with Google Apps Script?

```
function sendEmail() {
  var recipient = "example@gmail.com";
  var subject = "Test Email";
  var body = "This is a test email sent with Google Apps Script!";
  MailApp.sendEmail(recipient, subject, body);
}
```

Explanation: The sendEmail() function uses the MailApp class to send an email. The recipient's email address, subject, and body of the email are defined as variables and passed as parameters to the sendEmail() method.

How can I set a cell value in Google Sheets using Google Apps Script?

```
function setCellValue() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet1');
  sheet.getRange('A1').setValue('Hello, world!');
}
```

Explanation: The setCellValue() function uses the getActiveSpreadsheet() method to get a reference to the active spreadsheet, and getSheetByName() to get a reference to a specific sheet. Then, the setValue() method is used to set the value of the cell A1 to "Hello, world!".

How can I get the last row of a sheet in Google Sheets using Google Apps Script?

```
function getLastRow() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet1');
  var lastRow = sheet.getLastRow();
  Logger.log(lastRow);
}
```

Explanation: The getLastRow() function uses the getActiveSpreadsheet() method to get a reference to the active spreadsheet, and getSheetByName() to get a reference to a specific sheet. The getLastRow() method is used to get the index of the last row in the sheet.

How can I get the current date and time using Google Apps Script?

```
function getCurrentDateTime() {
  var now = new Date();
  Logger.log(now);
}
```

Explanation: The getCurrentDateTime() function creates a new Date object, which represents the current date and time. The Logger.log() method is used to print the date and time to the log.

How can I create a new sheet in a Google Sheet using Google Apps Script?

```
function createNewSheet() {
  var spreadsheet =
SpreadsheetApp.getActiveSpreadsheet();
  spreadsheet.insertSheet('New Sheet');
}
```

Explanation: The createNewSheet() function uses the getActiveSpreadsheet() method to get a reference to the active spreadsheet, and insertSheet() to create a new sheet with the specified name.

How can I delete a sheet in a Google Sheet using Google Apps Script?

```
function deleteSheet() {
  var spreadsheet =
SpreadsheetApp.getActiveSpreadsheet();
  var sheet = spreadsheet.getSheetByName('Sheet1');
  spreadsheet.deleteSheet(sheet);
}
```

Explanation: The deleteSheet() function uses the getActiveSpreadsheet() method to get a reference to the active spreadsheet, and getSheetByName() to get a reference to a specific sheet. The deleteSheet() method is used to delete the sheet.

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

```
function getValues() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet1');
  var range = sheet.getRange('A1:B10');
  var values = range.getValues();
  Logger.log(values);
}
```

Explanation: The getValues() function uses the getActiveSpreadsheet() method to get a reference to the active spreadsheet

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

```
function sortData() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet1');
  var range = sheet.getRange('A2:D11');
  range.sort([{column: 1, ascending: true}, {column: 2, ascending: true}]);
}
```

Explanation: The sortData() function uses the getActiveSpreadsheet() method to get a reference to the active spreadsheet, and getSheetByName() to get a reference to a specific sheet. The getRange() method is used to get a range of cells, and the sort() method is used to sort the data in ascending order based on two columns: column 1 and column 2.

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

```
function formatCells() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('S
heet1');
  var range = sheet.getRange('A2:D11');
  range.setBackground('#FFDAB9');
  range.setFontSize(14);
  range.setFontWeight('bold');
}
```

Explanation: The formatCells() function uses the getActiveSpreadsheet() method to get a reference to the active spreadsheet, and getSheetByName() to get a reference to a specific sheet. The getRange() method is used to get a range of cells, and the setBackground(), setFontSize(), and setFontWeight() methods are used to format the cells with a background color, font size, and font weight.

How can I use a custom function in Google Sheets using Google Apps Script?

```
function myFunction(input) {
  return input.toUpperCase();
}
```

Explanation: The myFunction() function is a custom function that can be used in Google Sheets. The function takes an input parameter and returns the uppercase version of the input. To use the function in a Google Sheet, simply enter "=myFunction(A1)" in a cell, where A1 is the cell with the input value.