

Apps Script Quiz 1



How can I create a custom menu in a Google Sheet using Google Apps Script?	2
How can I get the value of a cell in a Google Sheet using Google Apps Script?	3
How can I set the value of a cell in a Google Sheet using Google Apps Script?	3
How can I get the values of multiple cells in a Google Sheet using Google Apps Script?	4
How can I write data to a Google Sheet using Google Apps Script?	4
How can I search for a specific value in a Google Sheet using Google Apps Script?	5
How can I create a new sheet in a Google Sheet using Google Apps Script?	6
How can I delete a sheet in a Google Sheet using Google Apps Script?	6
How can I send an email using Google Apps Script?	7
How can I use a Google Sheets formula in Google Apps Script?	7

How can I create a custom menu in a Google Sheet using Google Apps Script?

```
function onOpen() {  
  var ui = SpreadsheetApp.getUi();  
  ui.createMenu('Custom Menu')  
    .addItem('Menu Item 1', 'menuItem1')  
    .addSeparator()  
    .addSubMenu(ui.createMenu('Submenu')  
      .addItem('Submenu Item 1', 'submenuItem1'))  
    .addToUi();  
}
```

```
function menuItem1() {  
  Browser.msgBox('You clicked Menu Item 1!');  
}
```

```
function submenuItem1() {  
  Browser.msgBox('You clicked Submenu Item 1!');  
}
```

Explanation: This code adds a custom menu to a Google Sheet when it is opened. The `onOpen()` function creates the custom menu, which has one menu item (Menu Item 1) and one submenu item (Submenu Item 1). When the user clicks on Menu Item 1, the `menuItem1()` function is called, which displays a message box. When the user clicks on Submenu Item 1, the `submenuItem1()` function is called, which also displays a message box.

How can I get the value of a cell in a Google Sheet using Google Apps Script?

```
function getCellValue() {  
  var sheet = SpreadsheetApp.getActiveSheet();  
  var cell = sheet.getRange("A1");  
  var value = cell.getValue();  
  Logger.log(value);  
}
```

Explanation: This code gets the value of cell A1 in the active sheet of a Google Sheet. The `getActiveSheet()` function returns the active sheet, the `getRange()` function returns a range of cells (in this case, cell A1), and the `getValue()` function returns the value of the cell. The value of the cell is logged to the execution log using the `Logger.log()` function.

How can I set the value of a cell in a Google Sheet using Google Apps Script?

```
function setCellValue() {  
  var sheet = SpreadsheetApp.getActiveSheet();  
  var cell = sheet.getRange("A1");  
  cell.setValue("Hello, World!");  
}
```

Explanation: This code sets the value of cell A1 in the active sheet of a Google Sheet to "Hello, World!". The `getActiveSheet()` function returns the active sheet, the `getRange()` function returns a range of cells (in this case, cell A1), and the `setValue()` function sets the value of the cell.

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

```
function getCellValues() {  
  var sheet = SpreadsheetApp.getActiveSheet();  
  var range = sheet.getRange("A1:B2");  
  var values = range.getValues();  
  Logger.log(values);  
}
```

Explanation: This code gets the values of cells A1 to B2 in the active sheet of a Google Sheet. The `getActiveSheet()` function returns the active sheet, the `getRange()` function returns a range of cells (in this case, cells A1 to B2), and the `getValues()` function returns a two-dimensional array of values.

How can I write data to a Google Sheet using Google Apps Script?

```
function writeToSheet() {  
  var sheet = SpreadsheetApp.getActiveSheet();  
  var values = [  
    ["John", "Doe", "johndoe@example.com"],  
    ["Jane", "Doe", "janedoe@example.com"]  
  ];  
  sheet.getRange(sheet.getLastRow() + 1, 1,  
values.length, values[0].length).setValues(values);  
}
```

Explanation This code writes data to the active sheet of a Google Sheet. The `getActiveSheet()` function returns the active sheet, and the `setValues()` function sets the values of a range of cells. In this example, the values are stored in a two-dimensional array and the `setValues()` function sets the values of a range starting from the last row of the sheet (using `getLastRow()` function) to the length of the values array in both rows and columns.

How can I search for a specific value in a Google Sheet using Google Apps Script?

```
function searchSheet() {
  var sheet = SpreadsheetApp.getActiveSheet();
  var range = sheet.getDataRange();
  var values = range.getValues();
  var searchValue = "John";
  for (var i = 0; i < values.length; i++) {
    for (var j = 0; j < values[i].length; j++) {
      if (values[i][j] == searchValue) {
        var cell = sheet.getRange(i+1, j+1);
        Logger.log(cell.getA1Notation());
      }
    }
  }
}
```

Explanation: This code searches for a specific value (`searchValue`) in the active sheet of a Google Sheet. The `getActiveSheet()` function returns the active sheet, the `getDataRange()` function returns the range of all data in the sheet, and the `getValues()` function returns a two-dimensional array of values. The code then

loops through each value in the array and checks if it matches the search value. If a match is found, the code gets the cell of the matching value using the `getRange()` function and logs the cell's A1 notation to the execution log using the `Logger.log()` function.

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

```
function createSheet() {  
  var ss = SpreadsheetApp.getActiveSpreadsheet();  
  var sheetName = "New Sheet";  
  var sheet = ss.insertSheet(sheetName);  
}
```

Explanation: This code creates a new sheet in the active Google Sheet. The `getActiveSpreadsheet()` function returns the active spreadsheet, and the `insertSheet()` function creates a new sheet with the specified name (`sheetName`) and returns the sheet object.

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

```
function deleteSheet() {  
  var ss = SpreadsheetApp.getActiveSpreadsheet();  
  var sheetName = "Sheet to delete";  
  var sheet = ss.getSheetByName(sheetName);  
  ss.deleteSheet(sheet);  
}
```

Explanation: This code deletes a sheet with a specific name (sheetName) from the active Google Sheet. The `getActiveSheet()` function returns the active spreadsheet, the `getSheetByName()` function returns the sheet with the specified name, and the `deleteSheet()` function deletes the sheet.

How can I send an email using Google Apps Script?

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

Explanation: This code sends an email using the MailApp service in Google Apps Script. The `sendEmail()` function takes three parameters: the email recipient, the email subject, and the email body.

How can I use a Google Sheets formula in Google Apps Script?

You can use Google Sheets formulas in Google Apps Script by using the `getRange()` function to get a range of cells, and then using the `setFormula()` function to set the formula for the range. Here's an example:

Suppose you have a Google Sheet with a table of data that includes a column of numbers in column B. You want to add a new column C that calculates the square of each number in column B using the `POWER()` function.

Here's how you can use a Google Sheets formula in Google Apps Script to accomplish this:

```
function addColumn() {  
  var sheet = SpreadsheetApp.getActiveSheet();  
  var range = sheet.getRange('C2:C');  
  range.setFormula('=POWER(B2,2)');  
}
```

Explanation:

The `getActiveSheet()` function returns the active sheet of the Google Sheet.

The `getRange()` function is used to select the range of cells where the formula should be applied. In this case, we want to apply the formula to column C starting at row 2 ('C2:C').

The `setFormula()` function is used to set the formula for the selected range. In this case, we're using the `POWER()` function to square the value in each cell in column B. The formula is written as a string in the function argument, and it must start with an equal sign (=) to indicate that it's a formula.

When you run this script, it will add a new column C to the sheet with the square of each value in column B. If you add new values to column B, the formula will automatically update to calculate the square of the new values.

