

Apps Script Questions with Solutions and Coding Examples 4



How can I create a new Google Slide using Google Apps Script?	2
How can I get the current date and time using Google Apps Script?	2
How can I add a menu to a Google Sheet using Google Apps Script?	3
How can I send an email from a Google Sheet using Google Apps Script?	4
How can I copy a Google Sheet using Google Apps Script?	5
How can I get data from a web API using Google Apps Script?	5
How can I read data from a Google Sheet using Google Apps Script?	6
How can I delete rows from a Google Sheet using Google Apps Script?	7
How can I get the URL of the current Google Sheet using Google Apps Script?	8
How can I add a menu to a Google Sheet using Google Apps Script?	8

How can I create a new Google Slide using Google Apps Script?

```
function createSlide() {  
  var presentation = SlidesApp.create('New  
Presentation');  
  var slide = presentation.getSlides()[0];  
  var shape =  
slide.insertShape(SlidesApp.ShapeType.RECTANGLE);  
  
shape.setWidth(200).setHeight(200).setTop(100).setLeft(  
100);  
  Logger.log('Presentation URL: ' +  
presentation.getUrl());  
}
```

Explanation: This code creates a new Google Slide using the `create()` function of the Slides service. The `create()` function takes a single argument, which is the title of the new presentation ('New Presentation'). The `getSlides()` function is used to get an array of slide objects, and the `[0]` index is used to select the first slide in the array. The `insertShape()` function is used to insert a rectangle shape onto the slide, and the `setWidth()`, `setHeight()`, `setTop()`, and `setLeft()` functions are used to position and size the shape. Finally, the `getUrl()` function is used to get the URL of the new presentation, which is then logged to the execution log.

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

```
function getCurrentDateTime() {  
  var now = new Date();  
  Logger.log('Current Date and Time: ' +  
now.toLocaleString());  
}
```

Explanation: This code gets the current date and time using the Date() function, which creates a new Date object with the current date and time. The toLocaleString() function is used to convert the date object to a string representation of the date and time in the local timezone, which is then logged to the execution log.

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

```
function onOpen() {  
  var ui = SpreadsheetApp.getUi();  
  ui.createMenu('Custom Menu')  
    .addItem('Menu Item 1', 'menuItem1')  
    .addItem('Menu Item 2', 'menuItem2')  
    .addToUi();  
}
```

```
function menuItem1() {  
  Logger.log('Menu Item 1 selected');  
}
```

```
function menuItem2() {  
  Logger.log('Menu Item 2 selected');  
}
```

Explanation: This code adds a custom menu to a Google Sheet using the `createMenu()` function of the Spreadsheet service. The `createMenu()` function takes a single argument, which is the name of the new menu ('Custom Menu'). The `addItem()` function is used to add two menu items to the new menu, with the labels 'Menu Item 1'

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

```
function sendEmail() {  
    var email = 'recipient@example.com';  
    var subject = 'Test Email';  
    var body = 'This is a test email sent from Google  
Apps Script.';  
    MailApp.sendEmail(email, subject, body);  
    Logger.log('Email sent to ' + email);  
}
```

Explanation: This code sends an email using the `sendEmail()` function of the Mail service. The `sendEmail()` function takes three arguments: the recipient email address ('recipient@example.com'), the subject of the email ('Test Email'), and the body of the email ('This is a test email sent from Google Apps Script.'). Finally, the `Logger.log()` function is used to log a message to the execution log indicating that the email was sent.

How can I copy a Google Sheet using Google Apps Script?

```
function copySheet() {  
  var originalSheet =  
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Original Sheet');  
  var destinationSheet =  
originalSheet.copyTo(SpreadsheetApp.getActiveSpreadsheet());  
  destinationSheet.setName('Copied Sheet');  
  Logger.log('Sheet copied to ' +  
destinationSheet.getName());  
}
```

Explanation: This code copies a Google Sheet using the `copyTo()` function of the Sheet object. The `copyTo()` function takes a single argument, which is the destination spreadsheet object (`SpreadsheetApp.getActiveSpreadsheet()` in this case). The `setName()` function is used to set the name of the copied sheet to 'Copied Sheet'. Finally, the `Logger.log()` function is used to log a message to the execution log indicating that the sheet was copied.

How can I get data from a web API using Google Apps Script?

```
function getApiData() {  
  var url = 'https://api.example.com/data';  
  var response = UrlFetchApp.fetch(url);  
}
```

```

    var data = JSON.parse(response.getContentText());
    Logger.log('Data: ' + JSON.stringify(data));
}

```

Explanation: This code retrieves data from a web API using the `fetch()` function of the `UrlFetch` service. The `fetch()` function takes a single argument, which is the URL of the API endpoint ('`https://api.example.com/data`' in this case). The `getContentText()` function is used to get the content of the response as a string, which is then parsed as JSON using the `JSON.parse()` function. Finally, the `Logger.log()` function is used to log a message to the execution log containing the data as a JSON string.

How can I read data from a Google Sheet using Google Apps Script?

```

function readData() {
    var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Data Sheet');
    var range = sheet.getDataRange();
    var values = range.getValues();
    for (var i = 0; i < values.length; i++) {
        Logger.log('Row ' + (i+1) + ': ' +
values[i].join(', '));
    }
}

```

Explanation: This code reads data from a Google Sheet using the `getDataRange()` and `getValues()` functions of the `Sheet` object. The `getDataRange()` function is used to get a range object

representing all the data in the sheet, and the `getValues()` function is used to get a 2D array of the values in the range. The for loop is used to iterate over each row of the array and log the values to the execution log.

How can I delete rows from a Google Sheet using Google Apps Script?

```
function deleteRows() {  
  var sheet =  
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Data Sheet');  
  var range = sheet.getDataRange();  
  var values = range.getValues();  
  for (var i = values.length - 1; i >= 0; i--) {  
    if (values[i][0] == 'delete') {  
      sheet.deleteRow(i+1);  
    }  
  }  
}
```

Explanation: This code deletes rows from a Google Sheet based on a condition using the `deleteRow()` function of the Sheet object. The `getDataRange()` and `getValues()` functions are used to get a 2D array of the values in the sheet. The for loop is used to iterate over each row of the array, starting from the last row, and delete the row if the first column contains the string 'delete'.

How can I get the URL of the current Google Sheet using Google Apps Script?

```
function getSheetUrl() {  
    var sheet = SpreadsheetApp.getActiveSpreadsheet();  
    var url = sheet.getUrl();  
    Logger.log('Sheet URL: ' + url);  
}
```

Explanation: This code gets the URL of the current Google Sheet using the `getUrl()` function of the Spreadsheet object. The `getActiveSpreadsheet()` function is used to get the currently active spreadsheet object. Finally, the `Logger.log()` function is used to log a message to the execution log containing the sheet URL.

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

```
function onOpen() {  
    var ui = SpreadsheetApp.getUi();  
    ui.createMenu('Custom Menu')  
        .addItem('Menu Item 1', 'menuItem1')  
        .addItem('Menu Item 2', 'menuItem2')  
        .addToUi();  
}
```

```
function menuItem1() {  
    // Do something  
}
```



```
function menuItem2() {  
  // Do something else  
}
```

Explanation: This code adds a custom menu to a Google Sheet using the `createMenu()` and `addItem()` functions of the `Ui` service. The `onOpen()` function is a reserved function that runs automatically when the sheet is opened. The `createMenu()` function is used to create a new menu with the name 'Custom Menu'. The `addItem()` function is used to add two menu items to the menu with the labels 'Menu Item 1' and 'Menu Item 2', and the functions 'menuItem1' and 'menuItem2' as their respective actions. Finally, the `addToUi()` function is used to add the menu to the UI of the sheet.