# Apps Script Questions with Solutions and Coding Examples 5

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

```
function formatCells() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1:C5');
  range.setBackground('#ffffff');
  range.setFontSize(12);
  range.setHorizontalAlignment('center');
  range.setVerticalAlignment('middle');
}
```

Explanation: This code formats cells in a Google Sheet using various formatting functions of the Range object. The getRange() function is used to get a range object representing cells A1:C5. The setBackground() function is used to set the background color of the cells to white. The setFontSize() function is used to set the font size of the cells to 12. The setHorizontalAlignment() function is used to set the horizontal alignment of the cells to center, and the setVerticalAlignment() function is used to set the vertical alignment of the cells to middle.

# How can I insert an image into a Google Sheet using Google Apps Script?

```
function insertImage() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var image =
UrlFetchApp.fetch('https://www.example.com/image.jpg');
  sheet.insertImage(image, 1, 1);
}
```

Explanation: This code inserts an image into a Google Sheet using the insertImage() function of the Sheet object. The getSheetByName() function is used to get a reference to the sheet where the image should be inserted. The UrlFetchApp.fetch() function is used to fetch the image from a URL and store it in the image variable. Finally, the insertImage() function is used to insert the image into cell A1 of the sheet.

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

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

Explanation: This code adds a new sheet to a Google Sheet using the insertSheet() function of the Spreadsheet object. The getActiveSpreadsheet() function is used to get a reference to the currently active spreadsheet. The insertSheet() function is used to insert a new sheet with the name 'New Sheet'.

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

```
function getCellValue() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var value = sheet.getRange('A1').getValue();
  Logger.log('Cell value: ' + value);
}
```

Explanation: This code gets the value of a cell in a Google Sheet using the getValue() function of the Range object. The getSheetByName() function is used to get a reference to the sheet containing the cell. The getRange() function is used to get a range object representing cell A1. Finally, the getValue() function is used to get the value of the cell, which 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.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1');
  range.setValue('New Value');
}
```

Explanation: This code sets the value of a cell in a Google Sheet using the setValue() function of the Range object. The getSheetByName() function is used to get a reference to the sheet containing the cell. The getRange() function is used to get a range object representing cell A1. Finally, the setValue() function is used to set the value of the cell to 'New Value'.

# How can I get the last row in a Google Sheet using Google Apps Script?

```
function getLastRow() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var lastRow = sheet.getLastRow();
  Logger.log('Last row: ' + lastRow);
}
```

Explanation: This code gets the last row in a Google Sheet using the getLastRow() function of the Sheet object. The getSheetByName() function is used to get a reference to the sheet containing the data. Finally, the getLastRow() function is used to get the index of the last row with data in it, which is logged to the execution log using the Logger.log() function.

# How can I get the last column in a Google Sheet using Google Apps Script?

```
function getLastColumn() {
```

```
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var lastColumn = sheet.getLastColumn();
  Logger.log('Last column: ' + lastColumn);
}
```

Explanation: This code gets the last column in a Google Sheet using the getLastColumn() function of the Sheet object. The getSheetByName() function is used to get a reference to the sheet containing the data. Finally, the getLastColumn() function is used to get the index of the last column with data in it, which is logged to the execution log using the Logger.log() function.

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

```
function getRowValues() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var values = sheet.getRange('A1:D1').getValues()[0];
  Logger.log('Row values: ' + values);
}
```

Explanation: This code gets all the values in a row of a Google Sheet using the getValues() 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 row of data (in this case, row 1). The [0] at the end of the getValues() function call is used to extract the values from the array of arrays returned by the

function (since we only want one row of data). Finally, the values are logged to the execution log using the Logger.log() function.

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

```
function getColumnValues() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var values =
sheet.getRange('A1:A4').getValues().map(function(row) {
    return row[0];
  });
  Logger.log('Column values: ' + values);
}
```

Explanation: This code gets all the values in a column of a Google Sheet using the getValues() 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 column of data (in this case, column A). The map() function is used to extract the values from the array of arrays returned by the getValues() function (since we only want one column of data). Finally, the values are logged to the execution log using the Logger.log() function.

# How can I clear the contents of a range in a Google Sheet using Google Apps Script?

```
function clearRange() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1:D4');
  range.clearContent();
}
```

Explanation: This code clears the contents of a range in a Google Sheet using the clearContent() 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 cleared. Finally, the clearContent() function is used to clear the contents of the range.

# How can I format the background color of a range in a Google Sheet using Google Apps Script?

```
function setRangeBackgroundColor() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1:D4');
  range.setBackground('#ffffff'); // set to white
```

```
}
```

Explanation: This code formats the background color of a range in a Google Sheet using the setBackground() 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 setBackground() function is used to set the background color of the range to white.

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

```
function setRangeFontColor() {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName('D
ata Sheet');
  var range = sheet.getRange('A1:D4');
  range.setFontColor('#ff0000'); // set to red
}
```

Explanation: This code formats the font color of a range in a Google Sheet using the setFontColor() 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 setFontColor() function is used to set the font color of the range to red.