QUIZ 100

Mastering Google Apps Script: A Comprehensive 100-Question Quiz

Introduction to Google Apps Script Quiz.. 7

Why Take This Quiz?.. 9

Questions Overview... 9

Apps Script Quiz questions..15

Question 1: What is Google Apps Script?... 15

Question 2: How can you access Google Apps Script Editor?.......................16

Question 3: Which service allows you to send emails using Google Apps Script?...17

Question 4: What is the purpose of the onEdit() trigger in Google Apps Script?...17

Question 5: How do you create a custom menu in Google Sheets using Apps Script?...18

Question 6: What is the method to retrieve the active sheet in a Google Sheets document?... 19

Question 7: Which of the following is a valid trigger type in Google Apps Script?...19

Question 8: How do you access data from a Google Sheet using Apps Script? 20

Question 9: Is it possible to call external APIs from Google Apps Script?.... 21

Question 10: How do you publish a web app using Google Apps Script?.... 21

Question 11: How can you access the Logger class in Google Apps Script?. 22

Question 12: Which class is used to interact with Google Drive in Google Apps Script?..23
Question 13: Which method is used to get the value of a named range in Google Sheets?...23
Question 14: How do you create a new file in Google Drive using Apps Script?..24
Question 15: What is the purpose of Session.getActiveUser() in Google Apps Script?..25
Question 16: How can you protect a range in Google Sheets using Apps Script?..25
Question 17: Which of these is a valid way to format a range in Google Sheets via Apps Script?...26
Question 18: How do you create a time-driven trigger in Google Apps Script?..26
Question 19: Can you use Google Apps Script to modify the content of a Google Doc?.. 27
Question 20: How do you retrieve responses from a Google Form using Apps Script?..28
Question 21: What is the correct method to append a row to a Google Sheet using Apps Script?... 29
Question 22: Which method is used to create a new Google Docs document using Apps Script?...29
Question 23: How can you access a specific cell in a Google Sheet?........... 30
Question 24: In Google Apps Script, what is the purpose of the CacheService?...30
Question 25: What does the SpreadsheetApp.flush() method do?.............31
Question 26: How do you set a trigger to run a function daily at a specific time in Apps Script?..31
Question 27: What is the main use of the PropertiesService in Google Apps Script?..32
Question 28: Which object is used for creating and manipulating UI components in a Google Sheets sidebar?...33

Question 29: How can you get the URL of the current Spreadsheet in Google Apps Script?...33
Question 30: What is the standard way to log errors in Google Apps Script?.. 34
Question 31: How do you create a Google Calendar event using Google Apps Script?...34
Question 32: What method is used to send an HTTP POST request in Google Apps Script?...35
Question 33: Which object is used to interact with Google Slides in Apps Script?...36
Question 34: How do you add a custom function to a Google Sheets cell?. 36
Question 35: In Google Apps Script, what is EventObject commonly used for?... 37
Question 36: Which method would you use to fetch the last row with data in a specific column in Google Sheets?...37
Question 37: What is the primary use of ScriptProperties in Google Apps Script?...38
Question 38: How can you create a UI alert dialog in a Google Sheets using Apps Script?...39
Question 39: What service in Google Apps Script is used for creating and manipulating HTML content?.. 39
Question 40: How do you access script logs in the new Google Apps Script IDE?...40
Question 41: What is the method to permanently delete a file in Google Drive using Apps Script?.. 40
Question 42: How can you access the active user's email address in Google Apps Script?...41
Question 43: Which service is used to work with transactions in a Google Sheet using Apps Script?.. 41
Question 44: How do you change the background color of a cell in Google Sheets using Apps Script?..42
Question 45: What does the MailApp.sendEmail() function return in Google Apps Script?...43

Question 46: In Google Apps Script, what does the LockService provide?.. 43
Question 47: How can you create a new folder in Google Drive using Apps Script?..44
Question 48: What is the purpose of the Utilities service in Google Apps Script?..44
Question 49: How can you format a date in Google Apps Script?...............45
Question 50: How do you add a note to a cell in Google Sheets using Apps Script?..45
Question 51: How do you retrieve the body content of a Google Doc using Google Apps Script?...46
Question 52: What is the purpose of the triggerUid property in trigger event objects?.. 47
Question 53: How can you set the data validation for a cell in Google Sheets using Apps Script?..47
Question 54: In Google Apps Script, which class is used to create a user interface in a standalone HTML page?... 48
Question 55: What does the SpreadsheetApp.openById(id) method do?... 48
Question 56: How can you add an image to a Google Slide using Apps Script?..49
Question 57: What is the main use of the Class FileIterator in Google Apps Script?..49
Question 58: How can you access a specific sheet in a Google Sheets document by its name?.. 50
Question 59: In Google Apps Script, what is the onInstall() trigger used for?.. 51
Question 60: How can you programmatically create a new Google Form using Apps Script?...51
Question 61: What method is used to insert a new row in a Google Sheet at a specific position?.. 52
Question 62: How do you access the title of an active Google Sheets document using Apps Script?... 52
Question 63: In Google Apps Script, which service is used to send a custom

HTML email?..53
Question 64: How do you retrieve all the sheets in a Google Sheets document using Apps Script?..54
Question 65: What is the correct way to access a range using A1 notation in Google Apps Script?...54
Question 66: How can you add a new item to an existing drop-down list in a cell in Google Sheets?..55
Question 67: In Google Apps Script, what does the DocumentApp.openById(id) method do?..55
Question 68: What is the main use of the UrlFetchApp service in Google Apps Script?..56
Question 69: How do you programmatically create a copy of a Google Sheet using Apps Script?...56
Question 70: In Google Apps Script, how do you set a cell's number format in Google Sheets?...57
Question 71: How do you programmatically set the width of a column in Google Sheets using Apps Script?...58
Question 72: In Google Apps Script, which method is used to trigger a function at regular intervals?..58
Question 73: How can you retrieve a list of all files in a specific folder in Google Drive using Apps Script?...59
Question 74: What is the main purpose of the Session service in Google Apps Script?..59
Question 75: How do you add a custom menu item to the Google Sheets UI using Apps Script?..60
Question 76: In Google Apps Script, how do you programmatically create a new Google Docs document and return its URL?......................................60
Question 77: How do you send an HTTP GET request and parse the JSON response in Google Apps Script?...61
Question 78: In Google Apps Script, what is the method to programmatically add a chart to a Google Sheets spreadsheet?................62
Question 79: How can you access the active cell in a Google Sheets spreadsheet using Apps Script?...62

Question 80: In Google Apps Script, how do you check if a file exists in Google Drive by name?.. 63
Question 81: How do you retrieve the URL of a file stored in Google Drive using Google Apps Script?... 63
Question 82: In Google Apps Script, which class is used to interact with Google Contacts?..64
Question 83: How can you programmatically insert an image into a Google Doc using Apps Script?... 65
Question 84: What function in Google Apps Script is used to encode a string in base64 format?..65
Question 85: How do you create a new Google Calendar event with guests using Apps Script?...66
Question 86: In Google Apps Script, how do you get the number of rows in a sheet that contain data?...67
Question 87: How can you programmatically change the font color of text in a range in Google Sheets using Apps Script?..67
Question 88: What method in Google Apps Script is used to create a new label in Gmail?.. 68
Question 89: How do you programmatically access the description of a Google Drive folder using Apps Script?... 68
Question 90: In Google Apps Script, which method is used to execute a SQL query in Google BigQuery?..69
Question 91: How do you get the active range in a Google Sheets spreadsheet using Apps Script?..69
Question 92: What method is used to add a new slide to a Google Slides presentation using Apps Script?.. 70
Question 93: In Google Apps Script, how do you programmatically create a new folder in a specific directory in Google Drive?.................................... 70
Question 94: How can you set a cell value in Google Sheets using Apps Script based on its A1 notation?...71
Question 95: In Google Apps Script, what is used to programmatically format currency in a range in Google Sheets?..72
Question 96: How do you create a time-driven trigger that runs every

weekday using Google Apps Script?.. 72
Question 97: What method in Google Apps Script is used to send an email with an attachment from Google Drive?.. 73
Question 98: In Google Apps Script, how do you check if a sheet exists in a Google Sheets document?...74
Question 99: What is the correct way to append a table to a Google Doc using Google Apps Script?.. 74
Question 100: How do you set the background color of a range in Google Sheets using RGB values in Apps Script?.. 75

Introduction to Google Apps Script Quiz

Google Apps Script, a powerful tool in the arsenal of a modern developer, offers immense flexibility and integration within the Google Workspace. To help learners and professionals alike gauge their understanding and mastery of this versatile platform, we present a comprehensive 100-question quiz. This quiz covers a wide array of topics, from the basics of Apps Script to more advanced functionalities.

Quiz Overview

Getting Started with Google Apps Script

- Questions 1-10 focus on introductory concepts like accessing the Apps Script Editor, sending emails, and understanding basic triggers and custom menus.

Interacting with Google Workspace

- Questions 11-30 delve into interacting with various Google Workspace applications like Google Drive, Sheets, Docs, and Forms. These include handling data, creating files, and manipulating document content.

Advanced Scripting Techniques

- Questions 31-50 explore more advanced scripting techniques, including working with UI components, handling external APIs, and creating time-driven triggers.

Utilities and Services in Apps Script

- Questions 51-70 cover the use of various utilities and services like Logger, CacheService, PropertiesService, and how they can be used to enhance script functionality.

Complex Applications and Customizations

- Questions 71-90 are designed for advanced users, focusing on complex operations like manipulating Google Sheets, setting up automated triggers, and customizing Google Workspace applications through scripting.

Deep Dive into Specific Functionalities

- Questions 91-100 aim to test in-depth knowledge in specific functionalities, including interaction with Google Slides, BigQuery, encoding techniques, and sophisticated sheet manipulations.

Why Take This Quiz?

- For Learners: Whether you're a beginner or an intermediate user, this quiz provides a structured way to assess your learning progress.
- For Professionals: If you're already using Google Apps Script in your workflows, this quiz helps identify areas for further improvement and exploration.
- For Educators: This comprehensive set of questions can be a valuable resource for structuring lessons or assessing student proficiency.

The 100-question quiz on Google Apps Script is more than just a test; it's a journey through the capabilities of a powerful scripting tool. By engaging with these questions, users can not only assess their current understanding but also discover new features and possibilities within the Google Workspace. Whether you aim to automate routine tasks, create complex applications, or simply streamline your workflows, mastering Google Apps Script is a step towards achieving these goals efficiently.

Questions Overview

Basic Concepts and Definitions

- What is Google Apps Script?

- What is the purpose of Session.getActiveUser() in Google Apps Script?
- What is the standard way to log errors in Google Apps Script?

Working with Google Sheets

- How do you set the background color of a range in Google Sheets using RGB values in Apps Script?
- Which method is used to get the value of a named range in Google Sheets?
- How can you protect a range in Google Sheets using Apps Script?
- Which of these is a valid way to format a range in Google Sheets via Apps Script?
- What is the correct method to append a row to a Google Sheet using Apps Script?
- How can you access a specific cell in a Google Sheet?
- How do you programmatically set the width of a column in Google Sheets using Apps Script?
- In Google Apps Script, how do you set a cell's number format in Google Sheets?
- How do you add a note to a cell in Google Sheets using Apps Script?
- How can you add a new item to an existing drop-down list in a cell in Google Sheets?
- What is the correct way to access a range using A1 notation in Google Apps Script?
- In Google Apps Script, what is used to programmatically format currency in a range in Google Sheets?

- How can you programmatically change the font color of text in a range in Google Sheets using Apps Script?
- How do you create a custom menu in Google Sheets using Apps Script?
- How do you access data from a Google Sheet using Apps Script?
- Which method would you use to fetch the last row with data in a specific column in Google Sheets?
- How do you retrieve all the sheets in a Google Sheets document using Apps Script?
- How can you access the active cell in a Google Sheets spreadsheet using Apps Script?
- How do you get the active range in a Google Sheets spreadsheet using Apps Script?
- In Google Apps Script, how do you check if a sheet exists in a Google Sheets document?
- How can you set the data validation for a cell in Google Sheets using Apps Script?

Interacting with Google Drive and Docs

- Which class is used to interact with Google Drive in Google Apps Script?
- How do you create a new file in Google Drive using Apps Script?
- What is the method to permanently delete a file in Google Drive using Apps Script?
- How can you create a new folder in Google Drive using Apps Script?
- How do you retrieve the URL of a file stored in Google Drive using Google Apps Script?

- How do you programmatically access the description of a Google Drive folder using Apps Script?
- How can you retrieve a list of all files in a specific folder in Google Drive using Apps Script?
- In Google Apps Script, how do you programmatically create a new folder in a specific directory in Google Drive?
- In Google Apps Script, how do you check if a file exists in Google Drive by name?
- Can you use Google Apps Script to modify the content of a Google Doc?
- How do you retrieve the body content of a Google Doc using Google Apps Script?
- Which method is used to create a new Google Docs document using Apps Script?
- In Google Apps Script, how do you programmatically create a new Google Docs document and return its URL?
- How can you programmatically insert an image into a Google Doc using Apps Script?
- What is the correct way to append a table to a Google Doc using Google Apps Script?

Working with Google Forms and Calendar

- How do you retrieve responses from a Google Form using Apps Script?
- How do you create a Google Calendar event using Google Apps Script?
- How do you create a new Google Calendar event with guests using Apps Script?

- How can you programmatically create a new Google Form using Apps Script?

Script Triggers and Events

- What is the purpose of the onEdit() trigger in Google Apps Script?
- How do you create a time-driven trigger in Google Apps Script?
- How do you set a trigger to run a function daily at a specific time in Apps Script?
- What is the main use of the Class FileIterator in Google Apps Script?
- In Google Apps Script, what is the onInstall() trigger used for?
- How do you create a time-driven trigger that runs every weekday using Google Apps Script?

Advanced Scripting and Services

- How can you access the Logger class in Google Apps Script?
- In Google Apps Script, what is the purpose of the CacheService?
- What does the SpreadsheetApp.flush() method do?
- What is the main use of the PropertiesService in Google Apps Script?
- Which object is used for creating and manipulating UI components in a Google Sheets sidebar?
- How can you get the URL of the current Spreadsheet in Google Apps Script?
- How do you publish a web app using Google Apps Script?
- Which service allows you to send emails using Google Apps Script?
- What method is used to send an HTTP POST request in Google Apps Script?

- Which object is used to interact with Google Slides in Apps Script?
- How do you add a custom function to a Google Sheets cell?
- In Google Apps Script, what is EventObject commonly used for?
- What is the primary use of ScriptProperties in Google Apps Script?
- How can you create a UI alert dialog in a Google Sheets using Apps Script?
- What service in Google Apps Script is used for creating and manipulating HTML content?
- How do you access script logs in the new Google Apps Script IDE?
- What method in Google Apps Script is used to create a new label in Gmail?
- How do you send an HTTP GET request and parse the JSON response in Google Apps Script?
- In Google Apps Script, what is the method to programmatically add a chart to a Google Sheets spreadsheet?
- Which of the following is a valid trigger type in Google Apps Script?
- What is the main purpose of the Session service in Google Apps Script?
- How do you add a custom menu item to the Google Sheets UI using Apps Script?
- What method is used to insert a new row in a Google Sheet at a specific position?
- How do you access the title of an active Google Sheets document using Apps Script?
- In Google Apps Script, which service is used to send a custom HTML email?

- What function in Google Apps Script is used to encode a string in base64 format?
- Is it possible to call external APIs from Google Apps Script?
- In Google Apps Script, which method is used to execute a SQL query in Google BigQuery?
- What method is used to add a new slide to a Google Slides presentation using Apps Script?
- What does the MailApp.sendEmail() function return in Google Apps Script?
- In Google Apps Script, what does the LockService provide?
- What is the purpose of the Utilities service in Google Apps Script?
- How can you format a date in Google Apps Script?
- In Google Apps Script, which class is used to interact with Google Contacts?
- What method in Google Apps Script is used to send an email with an attachment from Google Drive?
- How do you access the active user's email address in Google Apps Script?
- Which service is used to work with transactions in a Google Sheet using Apps Script?
- What is the purpose of the triggerUid property in trigger event objects?
- How can you access Google Apps Script Editor?

Apps Script Quiz questions

Question 1: What is Google Apps Script?

A) A Python-based scripting language

B) A JavaScript-based scripting language

C) A PHP-based web framework

D) A Ruby-based application platform

Answer: B

Explanation: Google Apps Script is a cloud-based scripting language for light-weight application development in the G Suite platform. It is based on JavaScript and provides a way to automate, integrate, and extend G Suite applications like Google Sheets, Docs, and Forms.

Question 2: How can you access Google Apps Script Editor?

A) From the Google Cloud Console

B) From the terminal using a special command

C) Directly from Google Docs, Sheets, or Forms

D) It can only be accessed via Google Drive

Answer: C

Explanation: Google Apps Script Editor can be accessed directly from Google Docs, Sheets, or Forms. You can open a script editor from these applications by clicking on “Extensions” > “Apps Script”.

Question 3: Which service allows you to send emails using Google Apps Script?

A) GmailApp

B) GoogleMail

C) EmailService

D) GoogleSendMail

Answer: A

Explanation and Code Sample: GmailApp is the service in Google Apps Script that allows you to send emails. Here's a basic example of sending an email:

```
function sendEmail() {
 GmailApp.sendEmail('recipient@example.com', 'Subject', 'Hello World!');
}
```

Question 4: What is the purpose of the onEdit() trigger in Google Apps Script?

A) To run a script every time a document is opened

B) To execute a script when a user edits a cell in a spreadsheet

C) To trigger a script when a file is uploaded to Google Drive

D) To schedule a script to run at specified intervals

Answer: B

Explanation: The onEdit() trigger is used to run a script automatically whenever a user edits a cell in a Google Sheets spreadsheet.

Question 5: How do you create a custom menu in Google Sheets using Apps Script?

A) Using the MenuApp service

B) Through the Sheets API

C) Using the addMenu method of the SpreadsheetApp service

D) It's not possible to create custom menus in Google Sheets

Answer: C

Explanation and Code Sample: You can create custom menus in Google Sheets using the addMenu method of the SpreadsheetApp service. Here's an example:

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

function menuItem1() {
 SpreadsheetApp.getUi().alert('You clicked the first menu item!');
}
```

Question 6: What is the method to retrieve the active sheet in a Google Sheets document?

A) getActiveSheet()

B) getCurrentSheet()

C) getSelectedSheet()

D) getOpenSheet()

Answer: A

Explanation: The getActiveSheet() method is used in Google Apps Script to retrieve the currently active sheet in a Google Sheets document.

Question 7: Which of the following is a valid trigger type in Google Apps Script?

A) onOpen

B) onChange

C) onEdit

D) All of the above

Answer: D

Explanation: All listed options (onOpen, onChange, onEdit) are valid trigger types in Google Apps Script. They are used to execute scripts based on specific actions like opening a document, editing a cell, or when changes are made to the document.

Question 8: How do you access data from a Google Sheet using Apps Script?

A) Using SQL queries

B) Through the SpreadsheetApp service

C) By utilizing the Google Sheets REST API

D) Through direct file system access

Answer: B

Explanation and Code Sample: You can access data from a Google Sheet using the SpreadsheetApp service. Here’s an example of how to read values from a sheet:

```
var sheet = SpreadsheetApp.getActiveSheet();
var data = sheet.getDataRange().getValues();
```

Question 9: Is it possible to call external APIs from Google Apps Script?

A) Yes, using the UrlFetchApp service

B) No, it's not possible to make external API calls

C) Only through Google Cloud Functions

D) Only if the API is part of G Suite

Answer: A

Explanation and Code Sample: Yes, it is possible to call external APIs from Google Apps Script using the UrlFetchApp service. Here's an example of making a GET request:

```
var response = UrlFetchApp.fetch('https://api.example.com/data');
var json = response.getContentText();
var data = JSON.parse(json);
```

Question 10: How do you publish a web app using Google Apps Script?

A) By deploying it through Google Cloud Platform

B) By using the “Deploy as web app” option in the script editor

C) Through the Google Play Store

D) It’s not possible to publish web apps using Google Apps Script

Answer: B

Explanation: You can publish a web app using Google Apps Script by clicking on “Deploy” > “New deployment” and then selecting “Web app” in the script editor. This allows you to create a URL that can be accessed by users.

Question 11: How can you access the Logger class in Google Apps Script?

A) LoggerApp.getLogger()

B) GoogleAppsScript.Logger

C) Simply Logger

D) AppScriptLogger.getService()

Answer: C

Explanation: In Google Apps Script, the Logger class can be accessed using Logger. This class is used for debugging purposes by logging messages which can be viewed in the Logs Viewer.

Question 12: Which class is used to interact with Google Drive in Google Apps Script?

A) DriveApp

B) GoogleDrive

C) DriveService

D) GoogleDriveApp

Answer: A

Explanation: DriveApp is the class used in Google Apps Script to interact with Google Drive. It allows for the manipulation of files and folders in the user's Google Drive.

Question 13: Which method is used to get the value of a named range in Google Sheets?

A) getNamedRange('name').getValue()

B) getRange('name').getValues()

C) getRangeByName('name').getValue()

D) getNamedRanges('name').getRange().getValues()

Answer: D

Explanation: To get the value of a named range, use getNamedRanges('name').getRange().getValues(). This method retrieves the range object for a named range and then gets its values.

Question 14: How do you create a new file in Google Drive using Apps Script?

A) DriveApp.createFile('New File', 'Hello, world!')

B) GoogleDrive.createFile('New File', 'Hello, world!')

C) DriveService.newFile('New File', 'Hello, world!')

D) FileApp.create('New File', 'Hello, world!')

Answer: A

Explanation and Code Sample: To create a new file in Google Drive using Google Apps Script, use DriveApp.createFile('New File', 'Hello, world!'). This method creates a new text file with the specified name and content.

```
var file = DriveApp.createFile('New File', 'Hello, world!');
```

Question 15: What is the purpose of Session.getActiveUser() in Google Apps Script?

A) To get the Google account email of the current user

B) To log the user out of the session

C) To get the active Google Workspace domain

D) To switch between different Google accounts

Answer: A

Explanation: Session.getActiveUser() returns the email address of the current user running the script. This can be useful for personalizing script actions based on the user.

Question 16: How can you protect a range in Google Sheets using Apps Script?

A) range.protect().setEditor('email@example.com')

B) range.setProtection(true)

C) SpreadsheetApp.Protect(range)

D) range.lock()

Answer: A

Explanation and Code Sample: To protect a range in Google Sheets, you use range.protect(). This method returns a protection object that can be modified, for example, to set specific editors.

```
var sheet = SpreadsheetApp.getActive().getSheetByName('Sheet1');
var range = sheet.getRange('A1:B10');
var protection = range.protect().setEditor('email@example.com');
```

Question 17: Which of these is a valid way to format a range in Google Sheets via Apps Script?

A) range.setStyle({fontSize: '12px', color: 'red'})

B) range.setFontSize(12).setFontColor('red')

C) range.applyStyle({fontSize: 12, color: 'red'})

D) SpreadsheetApp.formatRange(range, {fontSize: 12, color: 'red'})

Answer: B

Explanation: To format a range in Google Sheets using Apps Script, you use methods like setFontSize and setFontColor. These methods are chained directly to the range object.

Question 18: How do you create a time-driven trigger in Google Apps Script?

A)

ScriptApp.newTrigger('functionName').timeBased().everyHours(1).create()

B) TimeTrigger.create('functionName', 1)

C) ScriptApp.createTimeTrigger('functionName', 'hourly')

D) TriggerApp.newTimeTrigger('functionName').hourly()

Answer: A

Explanation and Code Sample: To create a time-driven trigger in Google Apps Script, use the ScriptApp.newTrigger function with a time-based builder method. For example, to create a trigger that runs every hour:

```
ScriptApp.newTrigger('functionName').timeBased().everyHours(1).create();
```

Question 19: Can you use Google Apps Script to modify the content of a Google Doc?

A) Yes, using the DocumentApp service

B) No, it's read-only for Google Docs

C) Only if the document is shared publicly

D) Only through a third-party API

Answer: A

Explanation: Yes, you can modify the content of a Google Doc using the

DocumentApp service in Google Apps Script. This service allows for reading, writing, and modifying the content of a Google Doc.

Question 20: How do you retrieve responses from a Google Form using Apps Script?

A) FormsApp.getResponses()

B) FormApp.getActiveForm().getResponses()

C) GoogleForms.getResponses('Form ID')

D) ResponsesService.getAllResponses()

Answer: B

Explanation and Code Sample: To retrieve responses from a Google Form, use FormApp.getActiveForm().getResponses(). This method fetches all responses submitted for the active form.

```
var form = FormApp.getActiveForm();
var responses = form.getResponses();
for (var i = 0; i < responses.length; i++) {
 // Process each response
}
```

Question 21: What is the correct method to append a row to a Google Sheet using Apps Script?

A) sheet.append('data1', 'data2', 'data3')

B) sheet.addRow(['data1', 'data2', 'data3'])

C) sheet.appendRow(['data1', 'data2', 'data3'])

D) sheet.insertRow(['data1', 'data2', 'data3'])

Answer: C

Explanation: sheet.appendRow(['data1', 'data2', 'data3']) is the correct method. This appends a new row to the bottom of the sheet with the specified values.

Question 22: Which method is used to create a new Google Docs document using Apps Script?

A) DocsApp.create('Title')

B) DocumentApp.create('Title')

C) GoogleDocs.createDocument('Title')

D) GDocs.newDocument('Title')

Answer: B

Explanation: DocumentApp.create('Title') is the correct method to create a new Google Docs document with the specified title.

Question 23: How can you access a specific cell in a Google Sheet?

A) sheet.getCell('A1')

B) sheet.getRange('A1')

C) sheet.selectCell('A1')

D) sheet.cell('A1')

Answer: B

Explanation: sheet.getRange('A1') is used to access a specific cell in a Google Sheet. The argument is the A1 notation of the desired cell.

Question 24: In Google Apps Script, what is the purpose of the CacheService?

A) To store data temporarily in your Google Drive

B) To cache responses from external API calls

C) To temporarily store data for faster access within the script

D) To backup script data in case of errors

Answer: C

Explanation: CacheService is used to temporarily store data for faster access within the script. This is particularly useful for reducing the number of calls to external services or databases.

Question 25: What does the SpreadsheetApp.flush() method do?

A) Clears all data from the spreadsheet

B) Forces the execution of all pending Spreadsheet changes

C) Refreshes the spreadsheet interface

D) Deletes all scripts associated with the spreadsheet

Answer: B

Explanation: SpreadsheetApp.flush() forces the execution of all pending Spreadsheet changes. It ensures that all previous Spreadsheet operations are completed before the script continues.

Question 26: How do you set a trigger to run a function daily at a specific time in Apps Script?

A)

ScriptApp.newTrigger('functionName').timeBased().atHour(10).everyDays(1).create()

B)

ScriptApp.newTrigger('functionName').timeBased().dailyAtHour(10).create()

C)

ScriptApp.newTrigger('functionName').timeBased().everyDayAt(10).create()

D)

ScriptApp.newTrigger('functionName').timeBased().daily().atHour(10).create()

Answer: B

Explanation:

ScriptApp.newTrigger('functionName').timeBased().dailyAtHour(10).create() is the correct method to set a trigger to run a function daily at a specific hour.

Question 27: What is the main use of the PropertiesService in Google Apps Script?

A) To store and retrieve global script properties

B) To modify properties of G Suite documents

C) To manage user settings and preferences

D) To configure script execution properties

Answer: A

Explanation: PropertiesService is used to store and retrieve properties that are specific to the script, regardless of the user. It can manage script, user, and document properties.

Question 28: Which object is used for creating and manipulating UI components in a Google Sheets sidebar?

A) UiApp

B) HtmlService

C) SidebarApp

D) InterfaceBuilder

Answer: B

Explanation: HtmlService is used for creating and manipulating UI components in a Google Sheets sidebar. It allows for the creation of HTML-based user interfaces, such as custom sidebars.

Question 29: How can you get the URL of the current Spreadsheet in Google Apps Script?

A) SpreadsheetApp.getActiveSpreadsheet().getUrl()

B) SpreadsheetApp.getCurrentSpreadsheet().link()

C) SpreadsheetApp.getActiveSheet().getLink()

D) SpreadsheetApp.getActive().getURL()

Answer: A

Explanation: SpreadsheetApp.getActiveSpreadsheet().getUrl() is the correct method to get the URL of the currently active Spreadsheet.

Question 30: What is the standard way to log errors in Google Apps Script?

A) Logger.logError('Error message')

B) Console.error('Error message')

C) Logger.log('Error message', Logger.LogLevel.ERROR)

D) Logger.log('Error message')

Answer: D

Explanation: In Google Apps Script, Logger.log('Error message') is used for logging, including errors. Although it doesn't differentiate error logs from regular logs, it's the standard way to log any messages, including errors.

Question 31: How do you create a Google Calendar event using Google Apps Script?

A) CalendarApp.createEvent(title, startTime, endTime)

B) GCalendar.newEvent(title, startTime, endTime)

C) GoogleCalendarApp.scheduleEvent(title, startTime, endTime)

D) CalendarService.addEvent(title, startTime, endTime)

Answer: A

Explanation: CalendarApp.createEvent(title, startTime, endTime) is the method used to create a new event in the user's default Google Calendar.

Question 32: What method is used to send an HTTP POST request in Google Apps Script?

A) UrlFetchApp.fetch(url, {method: 'post'})

B) HttpApp.post(url)

C) UrlService.sendPostRequest(url)

D) GoogleNet.post(url)

Answer: A

Explanation: UrlFetchApp.fetch(url, {method: 'post'}) is used to send an HTTP POST request. You can include additional options such as headers and payload in the second parameter.

Question 33: Which object is used to interact with Google Slides in Apps Script?

A) SlidesApp

B) PresentationApp

C) GoogleSlides

D) SlideService

Answer: A

Explanation: SlidesApp is the object used in Google Apps Script to interact with Google Slides, allowing for the creation, reading, and modification of presentations.

Question 34: How do you add a custom function to a Google Sheets cell?

A) Write the function in Apps Script and use the function name directly in the cell.

B) First register the function in Google Sheets, then use it in a cell.

C) Custom functions cannot be added directly to cells.

D) Use the =IMPORTFUNCTION() formula to import your custom function.

Answer: A

Explanation: Custom functions in Google Apps Script can be written and then used directly in Google Sheets cells by simply typing the function name, similar to built-in functions.

Question 35: In Google Apps Script, what is EventObject commonly used for?

A) Handling events in Google Forms

B) Tracking changes in Google Sheets

C) Managing browser-based events

D) Both A and B

Answer: D

Explanation: EventObject in Google Apps Script is commonly used for handling events in Google Forms and tracking changes in Google Sheets, such as in onEdit or onFormSubmit triggers.

Question 36: Which method would you use to fetch the last row with data in a specific column in Google Sheets?

A) sheet.getLastRow()

B) column.getLastRowWithData()

C) sheet.getDataRange().getLastRow()

D) range.endRowIndex()

Answer: A

Explanation: sheet.getLastRow() returns the number of the last row that contains data in the sheet. This is useful for appending data or analyzing the sheet's content.

Question 37: What is the primary use of ScriptProperties in Google Apps Script?

A) To store script-specific settings and data

B) To modify properties of the script editor

C) To share data between different scripts

D) To configure runtime properties of the script

Answer: A

Explanation: ScriptProperties is part of PropertiesService and is used to store script-specific settings and data. This data persists across script executions and is only accessible within the script.

Question 38: How can you create a UI alert dialog in a Google Sheets using Apps Script?

A) UiApp.createAlertDialog('Message')

B) SpreadsheetApp.getUi().alert('Message')

C) GoogleSheetsUI.alert('Message')

D) SheetUI.showAlert('Message')

Answer: B

Explanation: SpreadsheetApp.getUi().alert('Message') is used to create a simple UI alert dialog in Google Sheets. This method is part of the Ui class, which facilitates the creation of UI elements in Apps Script.

Question 39: What service in Google Apps Script is used for creating and manipulating HTML content?

A) HtmlService

B) WebApp

C) GoogleHtml

D) HtmlBuilder

Answer: A

Explanation: HtmlService is used to create and manipulate HTML content in Google Apps Script. It's commonly used for creating custom user interfaces for Google Workspace applications.

Question 40: How do you access script logs in the new Google Apps Script IDE?

A) View > Logs

B) Tools > Script editor > Logs

C) Execution > View logs

D) Script > Show logs

Answer: A

Explanation: In the new Google Apps Script IDE, script logs can be accessed by going to View > Logs. This allows you to see outputs from Logger.log() statements used in your script.

Question 41: What is the method to permanently delete a file in Google Drive using Apps Script?

A) DriveApp.getFileById('fileId').delete()

B) DriveApp.removeFile(file)

C) DriveApp.getFileById('fileId').deleteForever()

D) DriveApp.getFileById('fileId').setTrashed(true)

Answer: A

Explanation: DriveApp.getFileById('fileId').delete() is the correct method to permanently delete a file in Google Drive using Google Apps Script.

Question 42: How can you access the active user's email address in Google Apps Script?

A) Session.getActiveUser().getEmail()

B) UserApp.currentUser().getEmail()

C) GoogleUser.getEmail()

D) AppsScriptUser.email()

Answer: A

Explanation: Session.getActiveUser().getEmail() is used to get the email address of the current active user in Google Apps Script.

Question 43: Which service is used to work with transactions in a Google Sheet using Apps Script?

A) SpreadsheetApp

B) SheetTransactions

C) GoogleSheetsTransactions

D) There's no specific service for transactions

Answer: D

Explanation: In Google Apps Script, there's no specific service for handling transactions in a Google Sheet. Changes are made directly using the SpreadsheetApp service.

Question 44: How do you change the background color of a cell in Google Sheets using Apps Script?

A) range.setBackgroundColor('#FFFF00')

B) range.setStyle({backgroundColor: '#FFFF00'})

C) cell.color('#FFFF00')

D) range.setBackground('#FFFF00')

Answer: D

Explanation: range.setBackground('#FFFF00') is the method used to change the background color of a cell or range of cells in Google Sheets using Apps Script.

Question 45: What does the MailApp.sendEmail() function return in Google Apps Script?

A) The ID of the sent email

B) A boolean value indicating success or failure

C) Nothing, it's a void function

D) A GmailApp object

Answer: C

Explanation: MailApp.sendEmail() is a void function and does not return anything. It simply sends the email.

Question 46: In Google Apps Script, what does the LockService provide?

A) A way to password protect scripts

B) A service to encrypt data

C) A mechanism to prevent concurrent execution of certain sections of code

D) A service to lock files in Google Drive

Answer: C

Explanation: LockService provides a mechanism to prevent concurrent execution of certain sections of code in Google Apps Script, which is crucial in collaborative environments to avoid conflicts.

Question 47: How can you create a new folder in Google Drive using Apps Script?

A) DriveApp.createFolder('New Folder')

B) GDrive.newFolder('New Folder')

C) DriveApp.newFolder('New Folder')

D) GoogleDrive.createDirectory('New Folder')

Answer: A

Explanation: DriveApp.createFolder('New Folder') is the correct method for creating a new folder in Google Drive using Google Apps Script.

Question 48: What is the purpose of the Utilities service in Google Apps Script?

A) To provide utility methods for mathematics and text manipulation

B) To manage utilities like timers and counters

C) To configure the script settings and utilities

D) To access utility functions specific to Google Workspace

Answer: A

Explanation: The Utilities service in Google Apps Script provides utility methods for various tasks, including mathematics, text manipulation, date formatting, and encoding.

Question 49: How can you format a date in Google Apps Script?

A) DateApp.format('MM-dd-yyyy', date)

B) Utilities.formatDate(date, timeZone, format)

C) GoogleDate.format(date, 'MM-dd-yyyy')

D) DateFormat.format(date, 'MM-dd-yyyy')

Answer: B

Explanation: Utilities.formatDate(date, timeZone, format) is the correct method to format a date in Google Apps Script. This allows you to format the date according to a specific pattern.

Question 50: How do you add a note to a cell in Google Sheets using Apps Script?

A) cell.setNote('This is a note')

B) range.addNote('This is a note')

C) sheet.note('This is a note', 'A1')

D) range.setNote('This is a note')

Answer: D

Explanation: range.setNote('This is a note') is the method to add a note to a cell or a range of cells in Google Sheets using Apps Script.

Question 51: How do you retrieve the body content of a Google Doc using Google Apps Script?

A) DocumentApp.getActiveDocument().getBody()

B) DocsApp.openById('documentId').getContent()

C) GoogleDocs.getDocumentBody()

D) DocumentService.getActiveDocument().getBodyContent()

Answer: A

Explanation: DocumentApp.getActiveDocument().getBody() is the correct method to retrieve the body content of the active Google Doc.

Question 52: What is the purpose of the triggerUid property in trigger event objects?

A) To identify the user who created the trigger

B) To uniquely identify the trigger event

C) To track the number of times a trigger has fired

D) To determine the type of trigger (time-driven, edit, etc.)

Answer: B

Explanation: The triggerUid property in trigger event objects uniquely identifies the trigger event. This is useful for differentiating events in logs or for specific processing needs.

Question 53: How can you set the data validation for a cell in Google Sheets using Apps Script?

A) cell.setDataValidation(rule)

B) range.setDataValidator(validationRule)

C) range.setDataValidation(validationRule)

D) sheet.applyDataValidation('A1', rule)

Answer: C

Explanation: range.setDataValidation(validationRule) is the method used to set data validation for a range of cells in Google Sheets using Apps Script.

Question 54: In Google Apps Script, which class is used to create a user interface in a standalone HTML page?

A) HtmlService

B) UiApp

C) WebApp

D) PageApp

Answer: A

Explanation: HtmlService is used in Google Apps Script to create a user interface in a standalone HTML page. It allows the creation of HTML/CSS/JavaScript-based applications.

Question 55: What does the SpreadsheetApp.openById(id) method do?

A) Opens a spreadsheet in a new browser tab

B) Retrieves a Spreadsheet object by its ID

C) Edits a spreadsheet using its unique ID

D) Shares a spreadsheet using its ID

Answer: B

Explanation: SpreadsheetApp.openById(id) retrieves a Spreadsheet object by its unique ID, allowing you to programmatically interact with that spreadsheet.

Question 56: How can you add an image to a Google Slide using Apps Script?

A) SlidesApp.openById('slideId').addImage(url)

B) slide.insertImage(url)

C) SlidesApp.getActivePresentation().getSlide('slideId').insertImage(url)

D) slide.appendImage(url)

Answer: B

Explanation: slide.insertImage(url) is the correct method for adding an image to a specific slide in Google Slides using Google Apps Script.

Question 57: What is the main use of the Class FileIterator in Google Apps Script?

A) To iterate through a list of files in Google Drive

B) To create new files in a loop

C) To manage file downloads

D) To iterate through file contents for reading

Answer: A

Explanation: Class FileIterator is used in Google Apps Script to iterate through a list of files in Google Drive, typically obtained through a search or a specific query.

Question 58: How can you access a specific sheet in a Google Sheets document by its name?

A) SpreadsheetApp.getActiveSpreadsheet().getSheetByName('SheetName')

B) GoogleSheets.getSheet('SheetName')

C) SheetsApp.openByName('SheetName')

D) SpreadsheetApp.getSheet('SheetName')

Answer: A

Explanation: SpreadsheetApp.getActiveSpreadsheet().getSheetByName('SheetName') is the method to access a specific sheet in the active Google Sheets document by its name.

Question 59: In Google Apps Script, what is the onInstall() trigger used for?

A) To run code when the script is executed for the first time

B) To handle the installation process of add-ons

C) To initialize default settings upon script installation

D) To notify users when a new version of the script is installed

Answer: B

Explanation: The onInstall() trigger in Google Apps Script is used to handle the installation process of add-ons. It runs when an add-on is installed for the first time.

Question 60: How can you programmatically create a new Google Form using Apps Script?

A) FormsApp.create('Form Title')

B) GoogleForms.newForm('Form Title')

C) FormService.createForm('Form Title')

D) FormApp.create('Form Title')

Answer: D

Explanation: FormApp.create('Form Title') is the correct method to programmatically create a new Google Form with a specified title using Google Apps Script.

Question 61: What method is used to insert a new row in a Google Sheet at a specific position?

A) sheet.insertRowBefore(rowIndex)

B) sheet.addRow(rowIndex)

C) sheet.insertRow(rowIndex)

D) sheet.addNewRow(rowIndex)

Answer: A

Explanation: sheet.insertRowBefore(rowIndex) is the correct method to insert a new row in a Google Sheet at the specified position.

Question 62: How do you access the title of an active Google Sheets document using Apps Script?

A) SpreadsheetApp.getActiveSpreadsheet().getTitle()

B) SheetsApp.getActive().getName()

C) GoogleSheets.getActiveSpreadsheet().title()

D) SpreadsheetApp.getCurrent().getTitle()

Answer: A

Explanation: SpreadsheetApp.getActiveSpreadsheet().getTitle() is used to access the title of the active Google Sheets document.

Question 63: In Google Apps Script, which service is used to send a custom HTML email?

A) MailApp.sendHtmlEmail()

B) GmailApp.sendEmail({}, {}, {}, {htmlBody: htmlContent})

C) EmailApp.sendCustomEmail(htmlContent)

D) MailService.sendHtmlFormattedEmail()

Answer: B

Explanation: GmailApp.sendEmail({}, {}, {}, {htmlBody: htmlContent}) is used to send an email with custom HTML content. The htmlBody parameter in the options object allows you to specify the HTML content of the email.

Question 64: How do you retrieve all the sheets in a Google Sheets document using Apps Script?

A) SpreadsheetApp.getActiveSpreadsheet().getAllSheets()

B) SheetsApp.getSheets()

C) GoogleSheets.getActive().getSheets()

D) SpreadsheetApp.getCurrent().getEverySheet()

Answer: A

Explanation: SpreadsheetApp.getActiveSpreadsheet().getAllSheets() is the method to retrieve all the sheets in the active Google Sheets document.

Question 65: What is the correct way to access a range using A1 notation in Google Apps Script?

A) sheet.getRangeA1('A1:B2')

B) sheet.getRange('A1:B2')

C) range.select('A1:B2')

D) sheet.range('A1:B2')

Answer: B

Explanation: sheet.getRange('A1:B2') is used to access a specific range using A1 notation in Google Sheets through Apps Script.

Question 66: How can you add a new item to an existing drop-down list in a cell in Google Sheets?

A) Modify the cell's data validation rule to include the new item

B) Use a script to directly add an item to the drop-down list

C) Drop-down lists can't be modified after creation

D) Use a special addListItem method in Apps Script

Answer: A

Explanation: You would modify the cell's data validation rule to include the new item. Apps Script can programmatically change data validation rules, but there's no direct method to 'add' items to an existing drop-down list.

Question 67: In Google Apps Script, what does the DocumentApp.openById(id) method do?

A) Opens a Google Doc in a new browser tab

B) Retrieves a Document object by its ID

C) Edits a Google Doc using its unique ID

D) Shares a Google Doc using its ID

Answer: B

Explanation: DocumentApp.openById(id) retrieves a Document object by its unique ID, allowing you to programmatically interact with that Google Doc.

Question 68: What is the main use of the UrlFetchApp service in Google Apps Script?

A) To fetch resources and communicate with other hosts over the Internet

B) To create and manage URLs within a script

C) To track URL changes in a Google Sheet

D) To fetch URL parameters in web apps

Answer: A

Explanation: The UrlFetchApp service is used in Google Apps Script to fetch resources and communicate with other hosts over the Internet. It allows the script to make HTTP requests.

Question 69: How do you programmatically create a copy of a Google Sheet using Apps Script?

A) SpreadsheetApp.getActiveSpreadsheet().duplicate()

B) SheetsApp.cloneActiveSheet()

C) SpreadsheetApp.getActiveSpreadsheet().copy('New Copy')

D) GoogleSheets.copySpreadsheet()

Answer: C

Explanation: SpreadsheetApp.getActiveSpreadsheet().copy('New Copy') is the method to create a copy of the active Google Sheet and gives the new copy a specified name.

Question 70: In Google Apps Script, how do you set a cell's number format in Google Sheets?

A) range.setNumberFormat('0.00')

B) cell.formatNumber('0.00')

C) sheet.setCellFormat('A1', '0.00')

D) range.numberFormat('0.00')

Answer: A

Explanation: range.setNumberFormat('0.00') is the method to set the number format of a cell or range of cells in Google Sheets using Apps Script. This can format numbers, dates, or currencies.

Question 71: How do you programmatically set the width of a column in Google Sheets using Apps Script?

A) sheet.setColumnWidth(columnIndex, width)

B) column.setWidth(width)

C) range.setColumnWidth(width)

D) sheet.adjustColumnWidth(columnIndex, width)

Answer: A

Explanation: sheet.setColumnWidth(columnIndex, width) is the correct method to set the width of a specific column in Google Sheets using Apps Script.

Question 72: In Google Apps Script, which method is used to trigger a function at regular intervals?

A) ScriptApp.newTimeDrivenTrigger()

B) TimeTrigger.create()

C) ScriptApp.newTrigger().timeBased()

D) TriggerApp.setInterval()

Answer: C

Explanation: ScriptApp.newTrigger().timeBased() is used to create time-driven triggers in Google Apps Script. It allows you to specify intervals for triggering functions.

Question 73: How can you retrieve a list of all files in a specific folder in Google Drive using Apps Script?

A) DriveApp.getFolderById('folderId').getFiles()

B) DriveApp.retrieveFilesInFolder('folderId')

C) GoogleDrive.getFolderFiles('folderId')

D) FolderApp.listFiles('folderId')

Answer: A

Explanation: DriveApp.getFolderById('folderId').getFiles() is the method to retrieve a list of all files in a specific folder in Google Drive.

Question 74: What is the main purpose of the Session service in Google Apps Script?

A) To manage user sessions in web apps

B) To provide information about the current user's session

C) To create and maintain login sessions

D) To store session-specific data

Answer: B

Explanation: The Session service in Google Apps Script is primarily used to provide information about the current user's session, such as their email address and timezone.

Question 75: How do you add a custom menu item to the Google Sheets UI using Apps Script?

A) UiApp.createMenu('My Menu').addItem('Menu Item', 'functionName')

B) SpreadsheetApp.getUi().createMenu('My Menu').addItem('Menu Item', 'functionName').addToUi()

C) GoogleSheets.addMenu('My Menu', ['Menu Item'])

D) SheetUI.newMenu('My Menu').addItem('Menu Item')

Answer: B

Explanation: SpreadsheetApp.getUi().createMenu('My Menu').addItem('Menu Item', 'functionName').addToUi() is the correct method to add a custom menu item to the Google Sheets UI.

Question 76: In Google Apps Script, how do you programmatically create a new Google Docs document and return its URL?

A) DocsApp.create('Document Title').getUrl()

B) DocumentApp.create('Document Title').getUrl()

C) GoogleDocs.newDocument('Document Title').getLink()

D) DocumentService.createDoc('Document Title').getURL()

Answer: B

Explanation: DocumentApp.create('Document Title').getUrl() is the method to create a new Google Docs document and retrieve its URL.

Question 77: How do you send an HTTP GET request and parse the JSON response in Google Apps Script?

A) var response = UrlFetchApp.fetch(url); var data = JSON.parse(response.getContentText());

B) var data = HttpApp.get(url).json();

C) var response = WebApp.fetchGetRequest(url); var data = response.json();

D) var data = UrlService.get(url).parseJSON();

Answer: A

Explanation: var response = UrlFetchApp.fetch(url); var data = JSON.parse(response.getContentText()); is the correct way to send an HTTP GET request and parse the JSON response.

Question 78: In Google Apps Script, what is the method to programmatically add a chart to a Google Sheets spreadsheet?

A) sheet.insertChart(chartBuilder)

B) spreadsheet.createChart(chart)

C) chart.addToSheet(sheet)

D) sheet.addChart(chartBuilder.build())

Answer: D

Explanation: sheet.addChart(chartBuilder.build()) is used to add a chart to a Google Sheets spreadsheet. You first create a chart with a ChartBuilder and then add it to the sheet.

Question 79: How can you access the active cell in a Google Sheets spreadsheet using Apps Script?

A) SpreadsheetApp.getActiveSpreadsheet().getActiveCell()

B) SpreadsheetApp.getCurrentCell()

C) SheetsApp.activeCell()

D) GoogleSheets.getActiveCell()

Answer: A

Explanation: SpreadsheetApp.getActiveSpreadsheet().getActiveCell() is the method to access the active cell in the currently active Google Sheets spreadsheet.

Question 80: In Google Apps Script, how do you check if a file exists in Google Drive by name?

A) DriveApp.getFilesByName('fileName').hasNext()

B) DriveApp.fileExists('fileName')

C) GoogleDrive.checkFile('fileName')

D) DriveFiles.exists('fileName')

Answer: A

Explanation: DriveApp.getFilesByName('fileName').hasNext() is the method to check if a file with a given name exists in Google Drive. This method returns true if there's at least one file with the specified name.

Question 81: How do you retrieve the URL of a file stored in Google Drive using Google Apps Script?

A) DriveApp.getFileById('fileId').getUrl()

B) GoogleDrive.getFile('fileId').getLink()

C) DriveFiles.getUrlById('fileId')

D) DriveApp.getFile('fileId').getURL()

Answer: A

Explanation: DriveApp.getFileById('fileId').getUrl() is the method used to retrieve the URL of a file stored in Google Drive.

Question 82: In Google Apps Script, which class is used to interact with Google Contacts?

A) ContactsApp

B) GoogleContacts

C) ContactsService

D) GContactsApp

Answer: A

Explanation: ContactsApp is the class in Google Apps Script that provides methods to interact with Google Contacts.

Question 83: How can you programmatically insert an image into a Google Doc using Apps Script?

A)

DocumentApp.getActiveDocument().getBody().appendImage(imageBlob)

B) DocsApp.insertImage('documentId', imageBlob)

C) GoogleDocsApp.addImageToDocument(imageBlob)

D) DocumentService.getCurrentDocument().addImage(imageBlob)

Answer: A

Explanation:

DocumentApp.getActiveDocument().getBody().appendImage(imageBlob) is the correct method to insert an image into the body of an active Google Document.

Question 84: What function in Google Apps Script is used to encode a string in base64 format?

A) Utilities.base64Encode(string)

B) Base64.encode(string)

C) GoogleScripts.encodeBase64(string)

D) ScriptUtilities.toBase64(string)

Answer: A

Explanation: Utilities.base64Encode(string) is used to encode a given string into base64 format in Google Apps Script.

Question 85: How do you create a new Google Calendar event with guests using Apps Script?

A) CalendarApp.createEvent(title, startTime, endTime, {guests: 'email1,email2'})

B) GCalendar.createEventWithGuests(title, startTime, endTime, ['email1', 'email2'])

C) GoogleCalendar.newEvent(title, startTime, endTime).addGuests(['email1', 'email2'])

D) CalendarApp.createEvent(title, startTime, endTime).inviteGuests(['email1', 'email2'])

Answer: A

Explanation: CalendarApp.createEvent(title, startTime, endTime, {guests: 'email1,email2'}) is used to create a new Google Calendar event and automatically invite guests by providing their email addresses in a comma-separated string.

Question 86: In Google Apps Script, how do you get the number of rows in a sheet that contain data?

A) sheet.getDataRange().getNumRows()

B) sheet.getActiveRange().rowCount()

C) sheet.getNumberOfRows()

D) GoogleSheets.getRowCount('sheetId')

Answer: A

Explanation: sheet.getDataRange().getNumRows() is the method to get the number of rows that contain data in a sheet.

Question 87: How can you programmatically change the font color of text in a range in Google Sheets using Apps Script?

A) range.setFontColor('#FF0000')

B) cells.changeFontColor('A1:B2', '#FF0000')

C) sheet.setFontColor('A1:B2', '#FF0000')

D) range.setTextColor('#FF0000')

Answer: A

Explanation: range.setFontColor('#FF0000') is used to change the font color of text in a specified range in Google Sheets.

Question 88: What method in Google Apps Script is used to create a new label in Gmail?

A) GmailApp.createLabel(labelName)

B) GmailService.newLabel(labelName)

C) GoogleMail.createLabel(labelName)

D) MailApp.addLabel(labelName)

Answer: A

Explanation: GmailApp.createLabel(labelName) is the correct method to create a new label in Gmail using Google Apps Script.

Question 89: How do you programmatically access the description of a Google Drive folder using Apps Script?

A) DriveApp.getFolderById('folderId').getDescription()

B) GoogleDrive.getFolderDescription('folderId')

C) DriveApp.retrieveFolderInfo('folderId').description

D) FolderApp.getFolder('folderId').getDescription()

Answer: A

Explanation: DriveApp.getFolderById('folderId').getDescription() is used to programmatically access the description of a Google Drive folder.

Question 90: In Google Apps Script, which method is used to execute a SQL query in Google BigQuery?

A) BigQueryApp.query(sql)

B) GoogleBigQuery.execute(sql)

C) BigQuery.Services.executeQuery(sql)

D) BigQuery.Jobs.query(projectId, sql)

Answer: D

Explanation: BigQuery.Jobs.query(projectId, sql) is the method used to execute a SQL query in Google BigQuery through Google Apps Script.

Question 91: How do you get the active range in a Google Sheets spreadsheet using Apps Script?

A) SpreadsheetApp.getActiveSpreadsheet().getActiveRange()

B) SpreadsheetApp.getActiveRange()

C) SheetsApp.getCurrentRange()

D) GoogleSheets.getActiveRange()

Answer: A

Explanation: SpreadsheetApp.getActiveSpreadsheet().getActiveRange() is used to get the active range in the currently active Google Sheets spreadsheet.

Question 92: What method is used to add a new slide to a Google Slides presentation using Apps Script?

A) SlidesApp.getActivePresentation().addSlide()

B) PresentationApp.getCurrent().newSlide()

C) GoogleSlides.addNewSlide()

D) SlidesApp.createSlide()

Answer: A

Explanation: SlidesApp.getActivePresentation().addSlide() is the method used to add a new slide to the currently active Google Slides presentation.

Question 93: In Google Apps Script, how do you programmatically create a new folder in a specific directory in Google Drive?

A) DriveApp.getFolderById('folderId').createFolder('New Folder')

B) DriveApp.createFolder('parentFolderId', 'New Folder')

C) GoogleDrive.newFolderInDirectory('folderId', 'New Folder')

D) FolderApp.createSubFolder('folderId', 'New Folder')

Answer: A

Explanation: DriveApp.getFolderById('folderId').createFolder('New Folder') is the method to create a new folder within a specific directory in Google Drive.

Question 94: How can you set a cell value in Google Sheets using Apps Script based on its A1 notation?

A) sheet.getRange('A1').setValue('Hello World')

B) SpreadsheetApp.setValue('A1', 'Hello World')

C) GoogleSheets.setCellValue('A1', 'Hello World')

D) cell.setValue('A1', 'Hello World')

Answer: A

Explanation: sheet.getRange('A1').setValue('Hello World') is the method to set the value of a cell in Google Sheets based on its A1 notation.

Question 95: In Google Apps Script, what is used to programmatically format currency in a range in Google Sheets?

A) range.setNumberFormat('$#,##0.00')

B) range.formatAsCurrency()

C) sheet.currencyFormat('A1:B2')

D) GoogleSheets.formatRangeAsCurrency('A1:B2')

Answer: A

Explanation: range.setNumberFormat('$#,##0.00') is used to apply a currency format to a range in Google Sheets.

Question 96: How do you create a time-driven trigger that runs every weekday using Google Apps Script?

A)

ScriptApp.newTrigger('myFunction').timeBased().atWeekday(ScriptApp.WeekDay.MONDAY).create()

B)

ScriptApp.newTrigger('myFunction').timeBased().everyWeekday().create()

C) TriggerApp.createWeekdayTrigger('myFunction')

D) ScriptApp.newTrigger('myFunction').timeBased().onWeekdays().create()

Answer: B

Explanation:

ScriptApp.newTrigger('myFunction').timeBased().everyWeekday().create() is the correct way to create a time-driven trigger that runs the specified function every weekday.

Question 97: What method in Google Apps Script is used to send an email with an attachment from Google Drive?

A) MailApp.sendEmail({to: emailAddress, subject: subject, body: body, attachments: [DriveApp.getFileById('fileId').getBlob()]})

B) GmailApp.sendEmailWithAttachment(emailAddress, subject, body, 'fileId')

C) EmailService.sendAttachment(emailAddress, subject, body, 'fileId')

D) DriveApp.getFileById('fileId').sendAsEmail(emailAddress, subject, body)

Answer: A

Explanation: MailApp.sendEmail({to: emailAddress, subject: subject, body: body, attachments: [DriveApp.getFileById('fileId').getBlob()]}) is the method used to send an email with an attachment from Google Drive.

Question 98: In Google Apps Script, how do you check if a sheet exists in a Google Sheets document?

A) SpreadsheetApp.getActiveSpreadsheet().hasSheet('SheetName')

B)

SpreadsheetApp.getActiveSpreadsheet().getSheetByName('SheetName') != null

C) SheetsApp.exists('SheetName')

D) GoogleSheets.sheetExists('SheetName')

Answer: B

Explanation:

SpreadsheetApp.getActiveSpreadsheet().getSheetByName('SheetName') != null is used to check if a sheet with a given name exists in the active Google Sheets document.

Question 99: What is the correct way to append a table to a Google Doc using Google Apps Script?

A) DocumentApp.getActiveDocument().getBody().appendTable([['Cell 1', 'Cell 2'], ['Cell 3', 'Cell 4']])

B) DocsApp.insertTable('documentId', [['Cell 1', 'Cell 2'], ['Cell 3', 'Cell 4']])

C) GoogleDocs.addTable([['Cell 1', 'Cell 2'], ['Cell 3', 'Cell 4']])

D) DocumentService.getTable().appendRow(['Cell 1', 'Cell 2', 'Cell 3', 'Cell 4'])

Answer: A

Explanation:

DocumentApp.getActiveDocument().getBody().appendTable([['Cell 1', 'Cell 2'], ['Cell 3', 'Cell 4']]) is the correct method to append a table to a Google Doc.

Question 100: How do you set the background color of a range in Google Sheets using RGB values in Apps Script?

A) range.setBackgroundRGB(255, 0, 0)

B) range.setBackgroundColor('rgb(255, 0, 0)')

C) sheet.setRangeColor('A1:B2', 255, 0, 0)

D) range.setBackground('rgb(255, 0, 0)')

Answer: D

Explanation: range.setBackground('rgb(255, 0, 0)') is the method to set the background color of a range using RGB values in Google Sheets.