



1. What is Google Apps Script primarily used for?

- A. Building Android apps
- B. Automating and extending Google Workspace apps
- C. Managing Google Cloud infrastructure
- D. Designing static HTML websites

Correct answer: B

Explanation: Google Apps Script is a JavaScript-based platform used to automate, integrate, and extend Google Workspace apps like Sheets, Docs, Gmail, Forms, and Drive. It's not meant for Android apps or full cloud infrastructure management.

2. Which language is used to write Google Apps Script?

- A. Python
- B. Java
- C. JavaScript
- D. Go

Correct answer: C

Explanation: Apps Script uses JavaScript (specifically a server-side implementation similar to Node.js V8). While it runs on Google's servers, the language syntax is JavaScript.

3. Where do you typically create and edit Google Apps Script projects for a bound script?

- A. Google Cloud Console
- B. Google Sheets / Docs / Forms → Extensions → Apps Script
- C. Google Drive desktop application
- D. Chrome DevTools

Correct answer: B

Explanation: Container-bound scripts are created from within a specific document (Sheet, Doc, Form, etc.) using **Extensions** → **Apps Script**. This ties the script to that file.

4. Which of the following is a valid way to log output for debugging in Apps Script?

- A. `Console.write()`
- B. `Logger.log()`
- C. `Print.log()`
- D. `System.out.println()`

Correct answer: B

Explanation: `Logger.log()` is the built-in logging function in Apps Script. The logs can be seen via **View** → **Logs** or in the execution logs in the new editor.

5. What does `SpreadsheetApp.getActiveSpreadsheet()` return?

- A. The currently selected worksheet
- B. The current user's Drive folder
- C. The active spreadsheet file
- D. A list of all spreadsheets in Drive

Correct answer: C

Explanation: `getActiveSpreadsheet()` returns the active **Spreadsheet** object (the entire file), not just a single sheet or Drive folder.

6. How do you get the active sheet in a Google Sheets-bound script?

- A. `SpreadsheetApp.getActiveSheet()`
- B. `SpreadsheetApp.getActiveSpreadsheet().getActiveSheet()`
- C. `SheetsApp.getActiveSheet()`
- D. `SpreadsheetApp.getSheet()`

Correct answer: B

Explanation: You first get the active spreadsheet, then call `getActiveSheet()` on that object. Option A is not a valid global method.

7. Which method would you use to set a value in cell A1 of the active sheet?

- A. `sheet.setValue("A1", "Hello")`
- B. `sheet.getRange("A1").setValue("Hello")`
- C. `sheet.range("A1").value("Hello")`
- D. `sheet.setCell("A1", "Hello")`

Correct answer: B

Explanation: To write data, you must first obtain a `Range` object using `getRange("A1")` and then call `setValue()` on that range.

8. What is a “container-bound” script?

- A. A script that is zipped and downloaded
- B. A script that can only run in the Script Editor
- C. A script attached to a specific file like a Sheet or Doc
- D. A script that uses external APIs

Correct answer: C

Explanation: Container-bound scripts are associated with a specific Google Workspace file (Sheet, Doc, Form, etc.) and are opened from that file.

9. What is a “standalone” script in Apps Script?

- A. A script created only by admins
- B. A script not associated with any particular Google file
- C. A script that only uses advanced services
- D. A script that cannot be shared

Correct answer: B

Explanation: Standalone scripts exist on their own in Google Drive and are not tied to a specific document. They can still access files and services via code.

10. Which service is used to send emails in Apps Script?

- A. MailApp or GmailApp
- B. EmailApp
- C. MessageApp
- D. InboxApp

Correct answer: A

Explanation: Apps Script has **MailApp** (simple email sending) and **GmailApp** (richer Gmail-specific functionality). There is no EmailApp or InboxApp.

11. Which of the following sends a simple email using MailApp?

- A. `MailApp.sendMail("to@example.com", "Subject", "Body");`
- B. `Gmail.send("to@example.com", "Subject", "Body");`
- C. `Mail.send("to@example.com", "Subject", "Body");`
- D. `GmailApp.mail("to@example.com", "Subject", "Body");`

Correct answer: A

Explanation: `MailApp.sendMail(recipient, subject, body)` is the correct method signature for sending a basic email.

12. Which Apps Script service is used to interact with Google Drive files and folders?

- A. DriveService
- B. DriveApp
- C. FileApp
- D. CloudApp

Correct answer: B

Explanation: **DriveApp** is the core service used to list, create, and manage files and folders in the user's Google Drive.

13. Which of the following is NOT a built-in service in Apps Script?

- A. SpreadsheetApp
- B. DocumentApp
- C. GmailApp
- D. FirebaseApp

Correct answer: D

Explanation: FirebaseApp is not a built-in service; you'd use `UrlFetchApp` or the Advanced APIs for Firebase. The others are standard services.

14. What does `ScriptApp.getUserTriggers()` return?

- A. All triggers in the project
- B. All installable triggers for the active user in this script
- C. All simple triggers in the project
- D. The active trigger only

Correct answer: B

Explanation: `getUserTriggers()` returns a list of installable triggers created by the current user for a specified document or form.

15. Which of these is a simple trigger?

- A. `onEdit(e)`
- B. `timeDrivenTrigger()`
- C. `onDriveChange(e)`
- D. `onSendMail(e)`

Correct answer: A

Explanation: `onEdit(e)` is a simple trigger that runs automatically when a user edits a spreadsheet. Time-driven triggers must be created manually and are not "simple" triggers in code.

16. What is a key limitation of simple triggers like `onOpen` and `onEdit`?

- A. They never run
- B. They cannot access SpreadsheetApp
- C. They cannot prompt for authorization and have restricted services
- D. They only run for admins

Correct answer: C

Explanation: Simple triggers run under the active user without showing an auth dialog, so they cannot access services that require additional authorization or certain user data.

17. How do you create a time-driven (installable) trigger in Apps Script?

- A. By naming the function `onTime()`
- B. In the script editor: Triggers → Add Trigger
- C. Using `TimeApp.createTrigger()`
- D. By editing the manifest manually

Correct answer: B

Explanation: While you can also create triggers programmatically with `ScriptApp.newTrigger()`, the standard UI method is via **Triggers** → **Add Trigger** in the editor. There is no `TimeApp`.

18. Which method is used to fetch data from an external API?

- A. `ExternalApp.fetch()`
- B. `HttpApp.call()`
- C. `UrlFetchApp.fetch()`
- D. `ApiApp.request()`

Correct answer: C

Explanation: `UrlFetchApp.fetch(url)` is the core method to make HTTP requests to external APIs from Apps Script.

19. What type of value does `getValues()` return for a range?

- A. A single String
- B. A two-dimensional array of values
- C. A one-dimensional array of values
- D. A JSON object

Correct answer: B

Explanation: `Range.getValues()` always returns a 2D array (array of rows, each row an array of cells), even for a single cell range.

20. Which method would you use to write multiple rows of data to a sheet efficiently?

- A. Loop with `setValue()` for each cell
- B. `setValues()` on a range matching the data array
- C. `appendRow()` for each row
- D. `writeAll()`

Correct answer: B

Explanation: `setValues()` writes a 2D array to a contiguous range in one operation, which is much more efficient than multiple calls.

21. What is the best way to avoid hitting Apps Script quotas with too many calls to SpreadsheetApp?

- A. Use `Utilities.sleep()`
- B. Use batch operations like `getValues()` / `setValues()`
- C. Only use `Logger.log()`
- D. Use `SpreadsheetApp.flush()` constantly

Correct answer: B

Explanation: Reducing the number of service calls by using batch operations is a key performance optimization and helps stay within quotas.

22. Where can you see execution logs and errors from Apps Script?

- A. In Gmail settings
- B. In Google Drive trash
- C. In the Script Editor's Executions / Logs
- D. In Chrome Downloads

Correct answer: C

Explanation: The Script Editor provides an **Executions** panel where you can see run history, logs, stack traces, and errors.

23. Which object does the `onEdit(e)` trigger receive?

- A. A simple string value
- B. An event object with details about the edit
- C. A boolean flag
- D. No argument

Correct answer: B

Explanation: Simple triggers like `onEdit(e)` receive an event object (`e`) that contains information such as the edited range, old value, and new value.

24. How would you access the active user's email in an Apps Script that has proper scopes?

- A. `Session.getActiveUser().getEmail()`
- B. `UserApp.getEmail()`
- C. `ScriptApp.getUserEmail()`
- D. `GmailApp.getActiveUser().getEmail()`

Correct answer: A

Explanation: `Session.getActiveUser().getEmail()` returns the effective user's email, subject to domain and security settings. `ScriptApp.getUserEmail()` is not valid.

25. Which method would you use to create a custom menu in Google Sheets?

- A. `SpreadsheetApp.addMenu()`
- B. `getUi().createMenu()`
- C. `SpreadsheetApp.getUi().createMenu()`
- D. `SpreadsheetApp.createCustomMenu()`

Correct answer: C

Explanation: You obtain the UI with `SpreadsheetApp.getUi()` and then call `createMenu(name)` on that object, followed by `.addItem()` and `.addToUi()`.

26. Where is a custom menu typically added in a Google Sheets UI?

- A. In the sidebar
- B. In the context menu
- C. In the main menu bar at the top
- D. In the sheet tab menu

Correct answer: C

Explanation: Custom menus appear in the top menu bar (e.g., next to Help, Extensions) after the script runs `menu.addToUi()`.

27. Which of the following can show HTML interfaces in Apps Script?

- A. `HtmlService`
- B. `UiApp`
- C. `ViewService`
- D. `PanelApp`

Correct answer: A

Explanation: `HtmlService` is used to create and serve custom HTML-based UIs, such as sidebars and dialogs.

28. How do you show a sidebar with custom HTML in Google Sheets?

- A. `SpreadsheetApp.showSidebar(html)`
- B. `SpreadsheetApp.getUi().showSidebar(htmlOutput)`
- C. `HtmlService.showSidebar(html)`
- D. `SpreadsheetApp.getSidebar(html)`

Correct answer: B

Explanation: You create an `HtmlOutput` via `HtmlService.createHtmlOutput(...)` and pass it to `SpreadsheetApp.getUi().showSidebar()`.

29. What is the default runtime for modern Apps Script projects?

- A. Rhino
- B. V8
- C. JVM
- D. Nashorn

Correct answer: B

Explanation: Apps Script now uses the V8 runtime by default, bringing modern JavaScript features like `let`, `const`, arrow functions, and template literals.

30. Which statement is true about the V8 runtime?

- A. It doesn't support arrow functions
- B. It allows use of `require()`
- C. It supports ES6+ features like `let`, `const`, and template literals
- D. It requires using TypeScript

Correct answer: C

Explanation: The V8 runtime supports modern JS features such as `let`, `const`, classes, and arrow functions. It does not natively support Node's `require()`.

31. How do you include a library in an Apps Script project?

- A. Add `import` at the top of your script
- B. Use NPM install
- C. In project settings: add the library by its script ID
- D. Upload a `.jar` file

Correct answer: C

Explanation: Apps Script libraries are linked using their script ID via **Project Settings** → **Libraries** (or Services panel in the old editor).

32. Which of the following is a valid way to get a range by row and column?

- A. `sheet.range(1, 1)`
- B. `sheet.getRange(1, 1)`
- C. `sheet.getCell(1, 1)`
- D. `sheet.getCoordinates(1, 1)`

Correct answer: B

Explanation: `getRange(row, column)` is the correct method for numeric indices. There is `getRange()`, not `range()`.

33. How can you append a new row at the bottom of the active sheet?

- A. `sheet.addRow(["A", "B", "C"])`
- B. `sheet.insertRow(["A", "B", "C"])`
- C. `sheet.appendRow(["A", "B", "C"])`
- D. `sheet.addRowAtEnd(["A", "B", "C"])`

Correct answer: C

Explanation: `appendRow()` adds a row after the last non-empty row in the sheet and fills it with the values from the provided array.

34. Which method can be used to clear both values and formats from a range?

- A. `range.clear()`
- B. `range.clearValues()`
- C. `range.clearFormats()`
- D. `range.reset()`

Correct answer: A

Explanation: `clear()` removes values, formatting, and notes. `clearValues()` and `clearFormats()` are more specific.

35. What is the maximum script execution time for a single run in consumer accounts (approx)?

- A. 30 seconds
- B. 1 minute
- C. 6 minutes
- D. 30 minutes

Correct answer: C

Explanation: For standard consumer accounts, a single Apps Script execution typically has a limit of around 6 minutes (varies by context and quotas).

36. How can you ensure you don't exceed execution time when processing large data sets?

- A. Turn off logging
- B. Use recursion without limits
- C. Process data in chunks across multiple executions
- D. Use more `Utilities.sleep()`

Correct answer: C

Explanation: A common pattern is to process data in chunks (e.g., 500 rows at a time) and use properties or triggers to resume later.

37. What is `PropertiesService` used for?

- A. Managing file permissions
- B. Storing small amounts of key-value data
- C. Editing cell properties
- D. Styling text in documents

Correct answer: B

Explanation: `PropertiesService` lets you store small key-value pairs at script, user, or document level, useful for configuration or state.

38. Which type of property is shared across all users of the script?

- A. User properties
- B. Script properties
- C. Document properties
- D. Session properties

Correct answer: B

Explanation: Script properties are global to the script project and shared across all users. User properties are per user; document properties are per document.

39. Which Apps Script service is used to manipulate Google Docs?

- A. DocumentApp
- B. DocsService
- C. TextApp
- D. WordApp

Correct answer: A

Explanation: `DocumentApp` allows you to open and edit Google Docs files and their contents.

40. How do you get the active Google Doc in a Docs-bound script?

- A. `DocumentApp.openById()`
- B. `DocumentApp.getActiveDocument()`
- C. `DocsApp.getActive()`
- D. `Document.getActive()`

Correct answer: B

Explanation: `DocumentApp.getActiveDocument()` returns the document to which the script is bound.

41. What method would you use to insert text at the end of a Google Doc?

- A. `doc.getBody().appendParagraph("Text")`
- B. `doc.write("Text")`
- C. `doc.addText("Text")`
- D. `doc.getBody().insertText("Text")`

Correct answer: A

Explanation: `appendParagraph()` (or `appendText()` on body elements) is used to add content to the end of the document body.

42. Which service do you use for custom HTTP responses in web apps?

- A. `UrlFetchApp`
- B. `HttpService`

- C. `ContentService`
- D. `NetworkApp`

Correct answer: C

Explanation: `ContentService` lets you create text/JSON responses for web apps using `doGet(e)` or `doPost(e)`.

43. What function name is required to handle GET requests for a published web app?

- A. `onGet()`
- B. `doGet(e)`
- C. `getRequest(e)`
- D. `handleGet()`

Correct answer: B

Explanation: `doGet(e)` is the entry point for HTTP GET requests in Apps Script web apps.

44. Which deployment type allows users to access your script via a URL?

- A. Add-on
- B. Web app
- C. Library
- D. API executable only

Correct answer: B

Explanation: Deploying as a **web app** creates a URL endpoint where users can access your script.

45. How do you access query parameters in a `doGet(e)` function?

- A. `e.params`
- B. `e.queryString`
- C. `e.parameter`
- D. `e.request`

Correct answer: C

Explanation: `e.parameter` holds a key–value map of query parameters (for single values), and `e.parameters` holds arrays of values.

46. Which scope level is needed if your script sends emails on behalf of the user?

- A. Drive read-only
- B. Gmail read-only
- C. Gmail send
- D. No scope needed

Correct answer: C

Explanation: Sending mail via GmailApp typically requires a scope that allows sending emails (e.g., <https://www.googleapis.com/auth/gmail.send> or broader Gmail scopes).

47. What is an add-on in Apps Script?

- A. A Chrome extension
- B. A packaged script that extends Gmail, Docs, Sheets, etc.
- C. A theme for Google Workspace
- D. A mobile app

Correct answer: B

Explanation: Add-ons are Apps Script solutions packaged with a manifest (and sometimes HTML UI) that extend Google Workspace apps for users.

48. Where is the script manifest stored?

- A. In a hidden sheet
- B. In a file named `appsscript.json`
- C. In the Drive root
- D. There is no manifest

Correct answer: B

Explanation: Every Apps Script project has an `appsscript.json` manifest file that defines settings, scopes, add-on configuration, etc.

49. Which Apps Script class is used for converting data to and from JSON?

- A. `JsonApp`
- B. `Utilities`
- C. `ContentService`
- D. `ScriptApp`

Correct answer: B

Explanation: `Utilities.jsonParse()` (legacy) and `JSON.parse()` /

`JSON.stringify()` are used, but `Utilities` also offers encoding helpers. In modern V8, `JSON` is standard.

50. What does `SpreadsheetApp.flush()` do?

- A. Deletes all logs
- B. Immediately writes pending changes to the spreadsheet
- C. Clears cell formats
- D. Refreshes formulas

Correct answer: B

Explanation: `flush()` forces all pending spreadsheet changes to be applied immediately, which can be important before further reads or external operations.

51. Which simple trigger runs when a Google Form is submitted?

- A. `onSubmit(e)`
- B. `onFormSubmit(e)`
- C. `formSubmit(e)`
- D. `onEdit(e)`

Correct answer: B

Explanation: For form submissions, the simple trigger function name is `onFormSubmit(e)` (or an installable “On form submit” trigger).

52. How do you access the responses from a Form submit event?

- A. `e.values` (array of form responses)
- B. `e.responses` (object)
- C. `e.form.getValues()`
- D. `e.range.getValues()`

Correct answer: A

Explanation: For sheet-based form-submit triggers, `e.values` usually holds an array of the submitted row values.

53. What is the best way to handle errors in Apps Script?

- A. Ignore them
- B. Wrap risky code in `try...catch` blocks

- C. Disable logging
- D. Break the script into multiple files

Correct answer: B

Explanation: `try...catch` allows you to gracefully handle errors, log them, send alerts, or retry operations instead of crashing the entire script.

54. Which method pauses script execution for a specific number of milliseconds?

- A. `Utilities.wait()`
- B. `Utilities.sleep()`
- C. `ScriptApp.delay()`
- D. `TimeApp.pause()`

Correct answer: B

Explanation: `Utilities.sleep(milliseconds)` pauses execution, useful in rare cases when respecting API rate limits.

55. How can you make code reusable and easier to test?

- A. Put everything in `onOpen`
- B. Write one giant function
- C. Break logic into smaller functions and helpers
- D. Use only global variables

Correct answer: C

Explanation: Modularizing your code into small, focused functions makes it more maintainable, testable, and reusable.

56. Which statement is true about quotas in Apps Script?

- A. They are unlimited
- B. They depend on account type (consumer vs Workspace)
- C. They reset every hour only
- D. They don't apply to Gmail

Correct answer: B

Explanation: Quotas (like execution time, email sending, `UrlFetch` calls) vary by account type and sometimes by domain configuration.

57. What does `DriveApp.getFileById(id)` return?

- A. A Folder object
- B. A File object
- C. A Blob object
- D. A URL string

Correct answer: B

Explanation: `getFileById()` returns a `File` object representing the file with that ID in Drive.

58. How can you create a new spreadsheet via Apps Script?

- A. `SpreadsheetApp.create("My Sheet")`
- B. `SpreadsheetApp.newSpreadsheet("My Sheet")`
- C. `DriveApp.createSpreadsheet("My Sheet")`
- D. `SheetsApp.createFile("My Sheet")`

Correct answer: A

Explanation: `SpreadsheetApp.create(name)` creates a new spreadsheet file in the user's Drive and returns a `Spreadsheet` object.

59. Which method returns all sheets in a spreadsheet?

- A. `ss.getSheets()`
- B. `ss.getAllSheets()`
- C. `ss.getTabs()`
- D. `ss.listSheets()`

Correct answer: A

Explanation: `getSheets()` returns an array of `Sheet` objects in the spreadsheet.

60. How do you rename a sheet?

- A. `sheet.rename("New Name")`
- B. `sheet.setName("New Name")`
- C. `sheet.changeName("New Name")`
- D. `sheet.name = "New Name"`

Correct answer: B

Explanation: `setName()` is the correct method to rename a sheet tab.

61. Which method retrieves a document by its ID?

- A. `DocumentApp.getId(id)`
- B. `DriveApp.getDocument(id)`
- C. `DocumentApp.openById(id)`
- D. `Docs.open(id)`

Correct answer: C

Explanation: `DocumentApp.openById(id)` opens a Doc file given its file ID and returns a `Document` object.

62. Which of these is a recommended way to structure advanced scripts?

- A. Put configuration values in constants at the top
- B. Hard-code everything inside loops
- C. Use random variable names
- D. Avoid comments

Correct answer: A

Explanation: Keeping configuration (IDs, sheet names, etc.) in top-level constants improves readability and maintainability.

63. What does `Session.getScriptTimeZone()` return?

- A. The user's device time zone
- B. The project's time zone setting
- C. GMT only
- D. A boolean

Correct answer: B

Explanation: `Session.getScriptTimeZone()` returns the time zone configured for the script project (seen in project settings).

64. How can you interact with Calendar events from Apps Script?

- A. Using `CalendarApp`
- B. Using `TimeApp`
- C. Using `EventApp`
- D. Using `ScheduleApp`

Correct answer: A

Explanation: `CalendarApp` provides methods to list, create, and update Google Calendar events.

65. Which function name acts as the entry point for POST requests in a web app?

- A. `onPost(e)`
- B. `postRequest(e)`
- C. `doPost(e)`
- D. `handlePost(e)`

Correct answer: C

Explanation: `doPost(e)` handles HTTP POST requests for Apps Script web apps.

66. How can you restrict access to a web app so only users in your domain can access it?

- A. Use `UrlFetchApp`
- B. Set deployment access to “Only in my domain”
- C. Add a password in the code
- D. You cannot restrict it

Correct answer: B

Explanation: When deploying as a web app, you can choose who has access (e.g., “Anyone in my organization” or similar domain-scoped restrictions).

67. Which method is used to open a Drive folder by ID?

- A. `DriveApp.getFolder(id)`
- B. `DriveApp.openFolder(id)`
- C. `DriveApp.getFolderById(id)`
- D. `DriveApp.folder(id)`

Correct answer: C

Explanation: `getFolderById()` returns a `Folder` object with the specified ID.

68. What is a common pattern to pass data from client-side HTML/JS to server-side Apps Script?

- A. Direct database connection
- B. `google.script.run.serverFunctionName(data)`
- C. `UrlFetchApp.fetch()` from client
- D. `DocumentApp.send()`

Correct answer: B

Explanation: Within HTML served by `HtmlService`, you use the `google.script.run` API to call server-side Apps Script functions asynchronously.

69. How do you return JSON from a web app endpoint?

- A. Use `ContentService.createTextOutput(JSON.stringify(obj)).setMimeType(ContentService.MimeType.JSON)`
- B. Use `UrlFetchApp.returnJson(obj)`
- C. Use `JSONApp.send(obj)`
- D. Use `Logger.log(JSON.stringify(obj))`

Correct answer: A

Explanation: `ContentService` must be used to create a text output, and you set the MIME type to JSON.

70. Which Apps Script service is used to manipulate Google Slides?

- A. `SlideApp`
- B. `SlidesApp`
- C. `PresentationApp`
- D. `DeckApp`

Correct answer: B

Explanation: `SlidesApp` is the service to access and edit Google Slides presentations.

71. How can you access the active slide in a Slides-bound script?

- A. `SlidesApp.getActiveSlide()`
- B. `SlidesApp.getActivePresentation().getCurrentSlide()`
- C. `SlidesApp.getSlide()`
- D. `SlidesApp.getActive().getSlide()`

Correct answer: B

Explanation: You first get the active presentation, then call `getCurrentSlide()` on it.

72. Which of the following can trigger an Apps Script function automatically?

- A. Editing a spreadsheet
- B. Opening a document
- C. Submitting a form
- D. All of the above

Correct answer: D

Explanation: Simple and installable triggers can run on open, edit, form submit, and other events.

73. Which statement about simple triggers is correct?

- A. They require manual installation each time
- B. They can be used as web app entry points
- C. They run automatically based on the event name (function name)
- D. They can prompt the user with OAuth dialog

Correct answer: C

Explanation: Simple triggers run automatically when the function name matches a known event name (e.g., `onOpen`, `onEdit`, `onInstall`).

74. How do you enable an Advanced Google Service like “Sheets API” inside Apps Script?

- A. Use `SpreadsheetApp` directly
- B. Enable it from “Services” in the Script Editor and in Google Cloud console
- C. No setup required
- D. Install a Chrome extension

Correct answer: B

Explanation: Advanced services require enabling in the script project and corresponding Google Cloud API to be turned on.

75. Which of these is a benefit of using Advanced Google Services?

- A. Higher quotas
- B. Direct access to the underlying REST APIs
- C. Ability to bypass authentication
- D. Offline script execution

Correct answer: B

Explanation: Advanced services provide thin JavaScript wrappers around the REST APIs, enabling more granular control than the basic Apps Script services.

76. How do you get the ID of a spreadsheet from within a script bound to it?

- A. `SpreadsheetApp.getId()`
- B. `SpreadsheetApp.getActiveSpreadsheet().getId()`
- C. `DriveApp.getActiveId()`
- D. `Session.getId()`

Correct answer: B

Explanation: You first get the active spreadsheet and then call `getId()` on that object.

77. What does `Range.getA1Notation()` return?

- A. The value in the cell
- B. The A1-style string for the range (e.g., "B2:C5")
- C. The sheet name
- D. A JSON object

Correct answer: B

Explanation: `getA1Notation()` returns the A1-style reference of the range (column letters + row numbers).

78. Which is the best way to prevent scripts from breaking when sheet names change?

- A. Hard-code sheet names in many places
- B. Use sheet indexes only
- C. Store sheet names in constants and refer to them from one place
- D. Use random sheet names

Correct answer: C

Explanation: Using constants (like `const SHEET_NAME = 'Data';`) ensures you only update the name in one location if it changes.

79. What would `e.range` represent in an `onEdit(e)` function?

- A. The entire sheet
- B. The edited cell(s) as a Range object

- C. The whole spreadsheet
- D. The last cell of the sheet

Correct answer: B

Explanation: `e.range` is the specific range that was edited, allowing you to respond only to certain cells or areas.

80. What does it mean if a script requires authorization?

- A. The script will never run
- B. The script wants permission to access restricted services or data
- C. The script is broken
- D. The script needs to be paid for

Correct answer: B

Explanation: When a script first uses a scoped service (Gmail, Drive, Calendar, etc.), the user must grant permission for the script to access that data.

81. Which method retrieves the currently active cell in the active sheet?

- A. `sheet.getCurrentCell()`
- B. `sheet.getActiveCell()`
- C. `sheet.getSelectedCell()`
- D. `sheet.getCellActive()`

Correct answer: B

Explanation: `getActiveCell()` returns the current active cell (where the user's selection is) in the sheet.

82. How do you sort a range by the first column ascending?

- A. `range.sort({column: 1, ascending: true})`
- B. `range.sort(1)`
- C. `range.order(1, true)`
- D. `range.sortByColumn(1, true)`

Correct answer: A

Explanation: In the modern API, you pass an object like `{column: 1, ascending: true}` to `sort()`.

83. Which method is used to get the number of rows in a sheet that contain content?

- A. `sheet.getRowCount()`
- B. `sheet.getMaxRows()`
- C. `sheet.getLastRow()`
- D. `sheet.getActiveRow()`

Correct answer: C

Explanation: `getLastRow()` returns the index of the last non-empty row.

84. How can you protect a range so only certain users can edit it?

- A. Use `ProtectionApp`
- B. `range.protect()` and set editors
- C. `range.lock()`
- D. `sheet.secureRange()`

Correct answer: B

Explanation: `Range.protect()` returns a `Protection` object that lets you configure who can edit that range.

85. What is the role of `Session.getActiveUserLocale()`?

- A. Returns the browser language
- B. Returns the script editor language
- C. Returns the user's locale (like `en_US`)
- D. Returns their time zone

Correct answer: C

Explanation: `getActiveUserLocale()` returns the locale string that may affect formatting (dates, numbers, etc.).

86. How do you create a blob from a string?

- A. `Utilities.newBlob("Hello")`
- B. `DriveApp.createBlob("Hello")`
- C. `ContentService.createBlob("Hello")`
- D. `BlobApp.fromString("Hello")`

Correct answer: A

Explanation: `Utilities.newBlob(data)` creates a `Blob` object from given data, suitable for file creation or HTTP responses.

87. How can you send an attachment with MailApp?

- A. `MailApp.sendMail(to, subject, body, {attachments: [blob]})`
- B. `MailApp.sendAttachment(to, subject, body, blob)`
- C. `MailApp.sendMailWithFile(to, blob)`
- D. `MailApp.attach(to, blob)`

Correct answer: A

Explanation: `sendMail()` accepts an options object, where you can specify `attachments` as an array of Blobs.

88. How do you copy a file in Google Drive?

- A. `file.copy()`
- B. `file.makeCopy()`
- C. `DriveApp.copyFile(file)`
- D. `file.duplicate()`

Correct answer: B

Explanation: `makeCopy()` on a `File` object creates a duplicate of that file in Drive.

89. Which method lets you set the background color of a range?

- A. `range.setColor("#ff0000")`
- B. `range.setBackground("#ff0000")`
- C. `range.setFill("#ff0000")`
- D. `range.setBG("#ff0000")`

Correct answer: B

Explanation: `setBackground()` sets the background color of the cells in the range using a hex string or named color.

90. How can you use Apps Script to respond to installation of an add-on?

- A. Implement `onInstall(e)`
- B. Implement `onAdd(e)`
- C. Implement `onSetup(e)`
- D. Implement `onLoad(e)`

Correct answer: A

Explanation: `onInstall(e)` runs when an add-on is installed and usually calls `onOpen(e)` to set up menus.

91. Which of the following is the correct way to get the current script's project key in older Apps Script?

- A. `ScriptApp.getProjectKey()`
- B. `ScriptApp.getId()`
- C. `ScriptApp.getProjectId()`
- D. `DriveApp.getProjectKey()`

Correct answer: A

Explanation: Historically, `getProjectKey()` returned the project key (deprecated for new editor; now project ID is mostly managed automatically). Still, it's the closest correct option here.

92. How can you schedule a script to run every hour?

- A. Name the function `onHour()`
- B. Create a time-driven trigger in Triggers and choose "Hour timer"
- C. Use `Utilities.sleep(3600000)` in a loop
- D. It's not possible

Correct answer: B

Explanation: Use an installable time-driven trigger configured to run hourly.

93. How do you read environment-like configuration from the script properties?

- A. `PropertiesService.getEnv()`
- B. `PropertiesService.getScriptProperties().getProperty("KEY")`
- C. `DriveApp.getProperties("KEY")`
- D. `ScriptApp.getConfig("KEY")`

Correct answer: B

Explanation: You use `PropertiesService.getScriptProperties()` to access script-level properties and then `getProperty(name)`.

94. Which method allows you to programmatically create a new trigger?

- A. `ScriptApp.createTrigger()`
- B. `ScriptApp.newTrigger(functionName)`
- C. `ScriptApp.trigger(functionName)`
- D. `ScriptApp.setTrigger(functionName)`

Correct answer: B

Explanation: `ScriptApp.newTrigger(functionName)` is used to define a new trigger, then you chain methods such as `.timeBased()` or `.forSpreadsheet()` and `.create()`.

95. How can you iterate over all files in a Drive folder?

- A. `folder.GetFiles()` returns a `FileIterator` that you loop with `.hasNext()` and `.next()`
- B. `folder.getAllFiles()` returns an array
- C. `DriveApp.listFiles(folder)`
- D. `folder.files` is an array

Correct answer: A

Explanation: `getFiles()` returns a `FileIterator`, which you use in a `while (files.hasNext())` loop to access each file.

96. How do you evaluate a custom function in Google Sheets written in Apps Script?

- A. Call it from the Script Editor
- B. Use it as a formula in a cell (e.g., `=MYFUNCTION(A1)`)
- C. Run it from Triggers
- D. It evaluates automatically without formulas

Correct answer: B

Explanation: Custom functions are called like built-in formulas in cells using `=FUNCTION_NAME(...)`.

97. Which is a limitation of custom functions in Sheets?

- A. They cannot read cell values
- B. They cannot access services that require authorization (like MailApp)
- C. They cannot use JavaScript
- D. They cannot return values

Correct answer: B

Explanation: Custom functions must be “pure” and cannot use services requiring authorization or modify other cells; they just compute and return values.

98. How do you test a function that is triggered by `onEdit(e)` without editing the sheet?

- A. You cannot test it
- B. Create a test function that calls `onEdit()` with a mock event object
- C. Use `ScriptApp.run(onEdit)`
- D. Just click Run and hope

Correct answer: B

Explanation: A common pattern is to build a wrapper test function that constructs a fake `e` object and passes it to `onEdit(e)`.

99. Which pattern helps you avoid duplicated code when working with multiple Sheets in the same spreadsheet?

- A. Copy-paste the same code everywhere
- B. Write helper functions that accept `sheet` as a parameter
- C. Use only one sheet
- D. Use global `sheet` everywhere

Correct answer: B

Explanation: Helper functions that accept a `Sheet` object allow you to reuse logic for different sheets simply by passing a different sheet.

100. What is the first thing to do when debugging a script that suddenly stops working?

- A. Delete the project
- B. Turn off all triggers
- C. Check the execution logs / executions history for errors
- D. Re-authorize all services

Correct answer: C

Explanation: Execution logs and history show error messages, line numbers, and stack traces, which are crucial starting points for debugging.