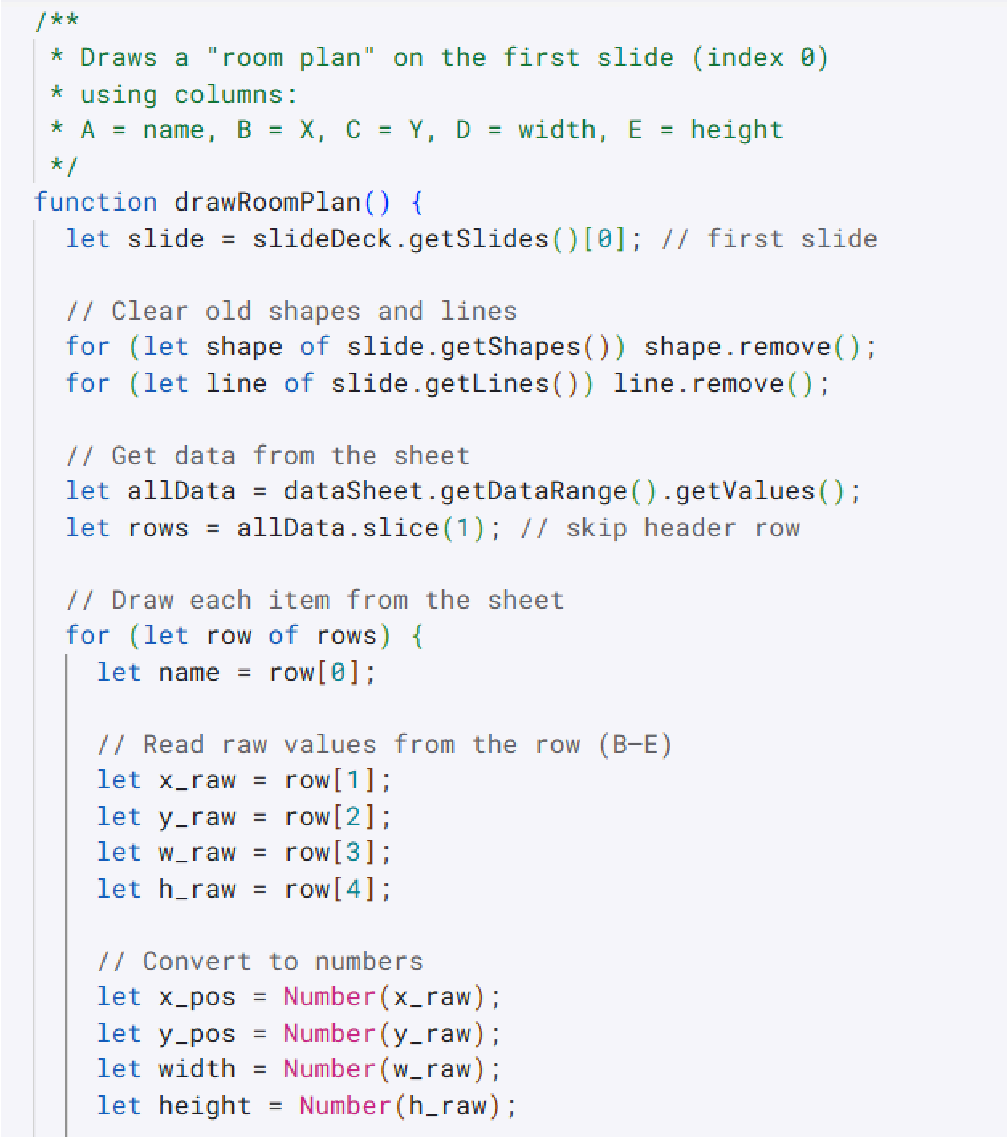
Homework Assignment – Automating data transfer in google sheets with Apps Script

Report by Riina Kikkas

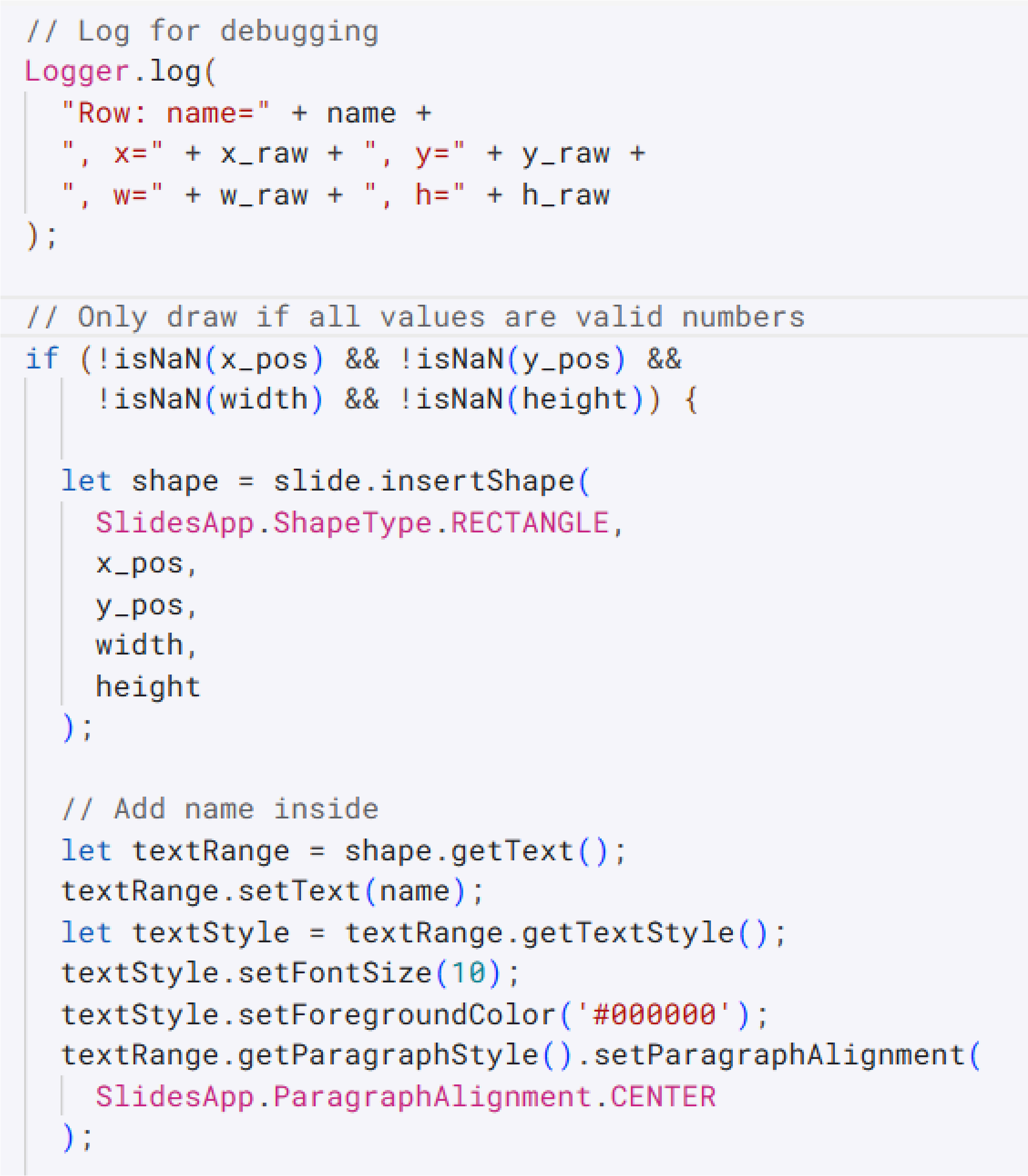
[Spreadsheet link]

Function: drawRoomPlan()

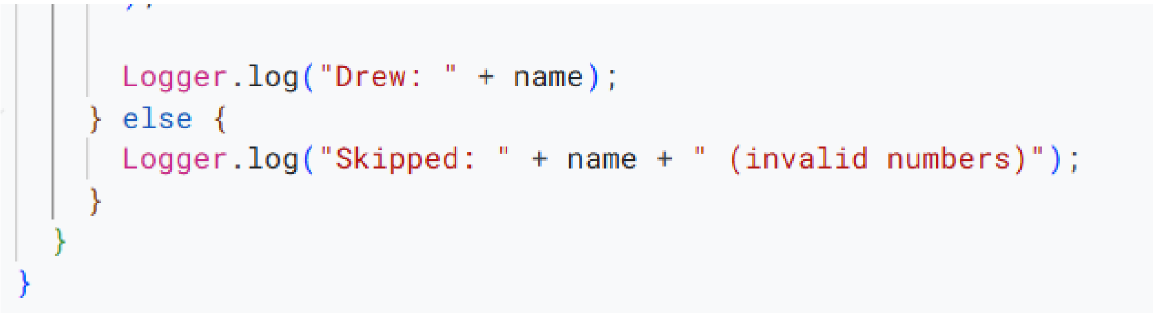
This function reads data from my Google Sheet and draws rectangles on the first slide of my Google Slides presentation.
Each row in the sheet represents one item (for example: Sofa, Table, TV, or Shelf).
The script takes the item name, X and Y positions, width, and height from the spreadsheet and uses them to place and size each shape.
It also writes the item’s name inside the rectangle and centers the text.
If a cell in the spreadsheet is empty or not a number, that row is skipped.
This function basically creates a simple room layout automatically using data from the spreadsheet.



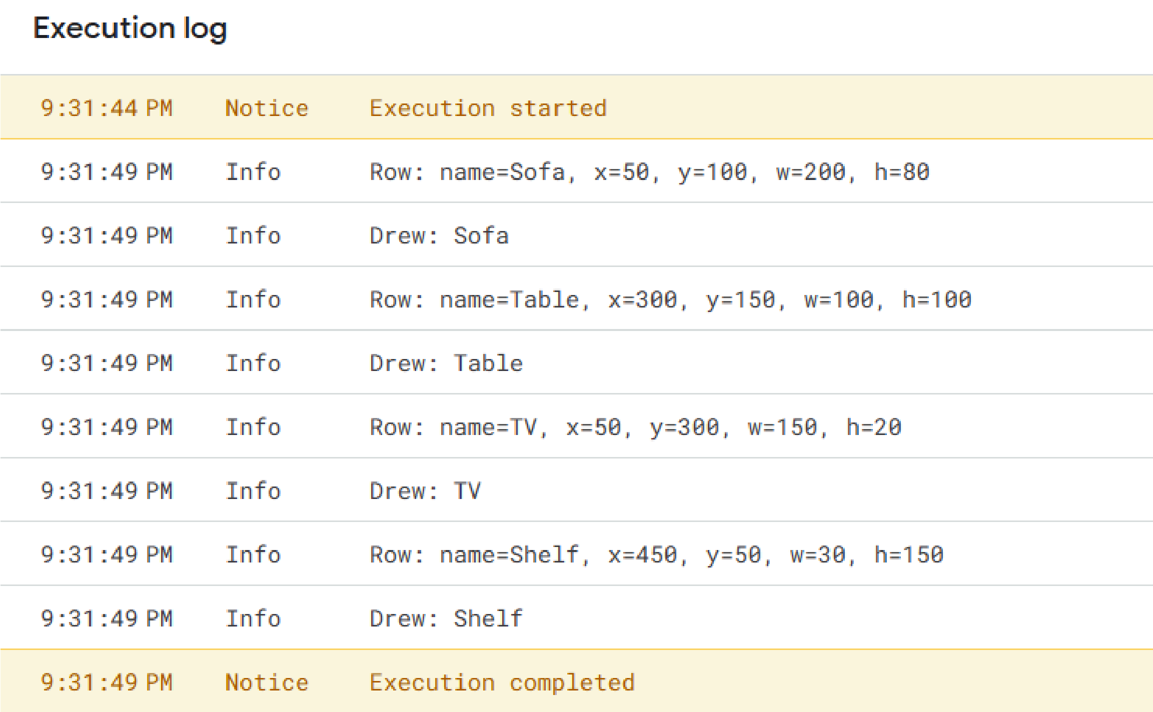
```
/**
 * Draws a "room plan" on the first slide (index 0)
 * using columns:
 * A = name, B = X, C = Y, D = width, E = height
 */
function drawRoomPlan() {
  let slide = slideDeck.getSlides()[0]; // first slide

  // Clear old shapes and lines
  for (let shape of slide.getShapes()) shape.remove();
  for (let line of slide.getLines()) line.remove();

  // Get data from the sheet
  let allData = dataSheet.getDataRange().getValues();
  let rows = allData.slice(1); // skip header row

  // Draw each item from the sheet
  for (let row of rows) {
    let name = row[0];

    // Read raw values from the row (B–E)
    let x_raw = row[1];
    let y_raw = row[2];
    let w_raw = row[3];
    let h_raw = row[4];

    // Convert to numbers
    let x_pos = Number(x_raw);
    let y_pos = Number(y_raw);
    let width = Number(w_raw);
    let height = Number(h_raw);
```



```
// Log for debugging
Logger.log(
  "Row: name=" + name +
  ", x=" + x_raw + ", y=" + y_raw +
  ", w=" + w_raw + ", h=" + h_raw
);

// Only draw if all values are valid numbers
if (!isNaN(x_pos) && !isNaN(y_pos) &&
    !isNaN(width) && !isNaN(height)) {

  let shape = slide.insertShape(
    SlidesApp.ShapeType.RECTANGLE,
    x_pos,
    y_pos,
    width,
    height
  );

  // Add name inside
  let textRange = shape.getText();
  textRange.setText(name);
  let textStyle = textRange.getTextStyle();
  textStyle.setFontSize(10);
  textStyle.setForegroundColor('#000000');
  textRange.getParagraphStyle().setParagraphAlignment(
    SlidesApp.ParagraphAlignment.CENTER
  );
```

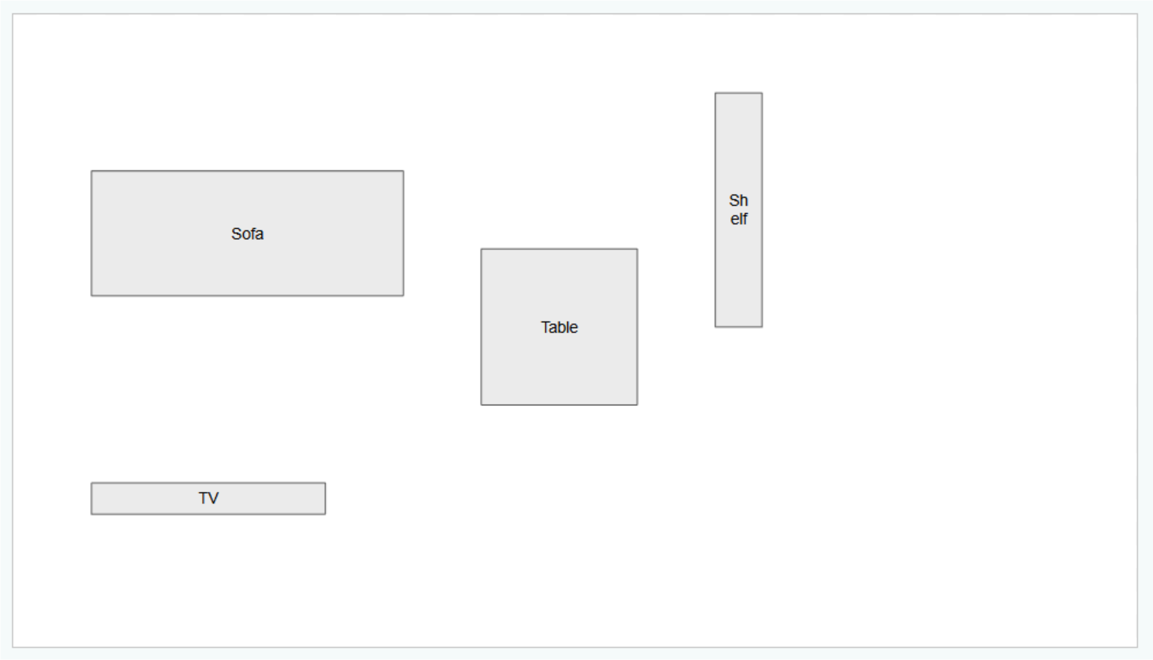


```

      Logger.log("Drew: " + name);
    } else {
      Logger.log("Skipped: " + name + " (invalid numbers)");
    }
  }
}
```



Execution log

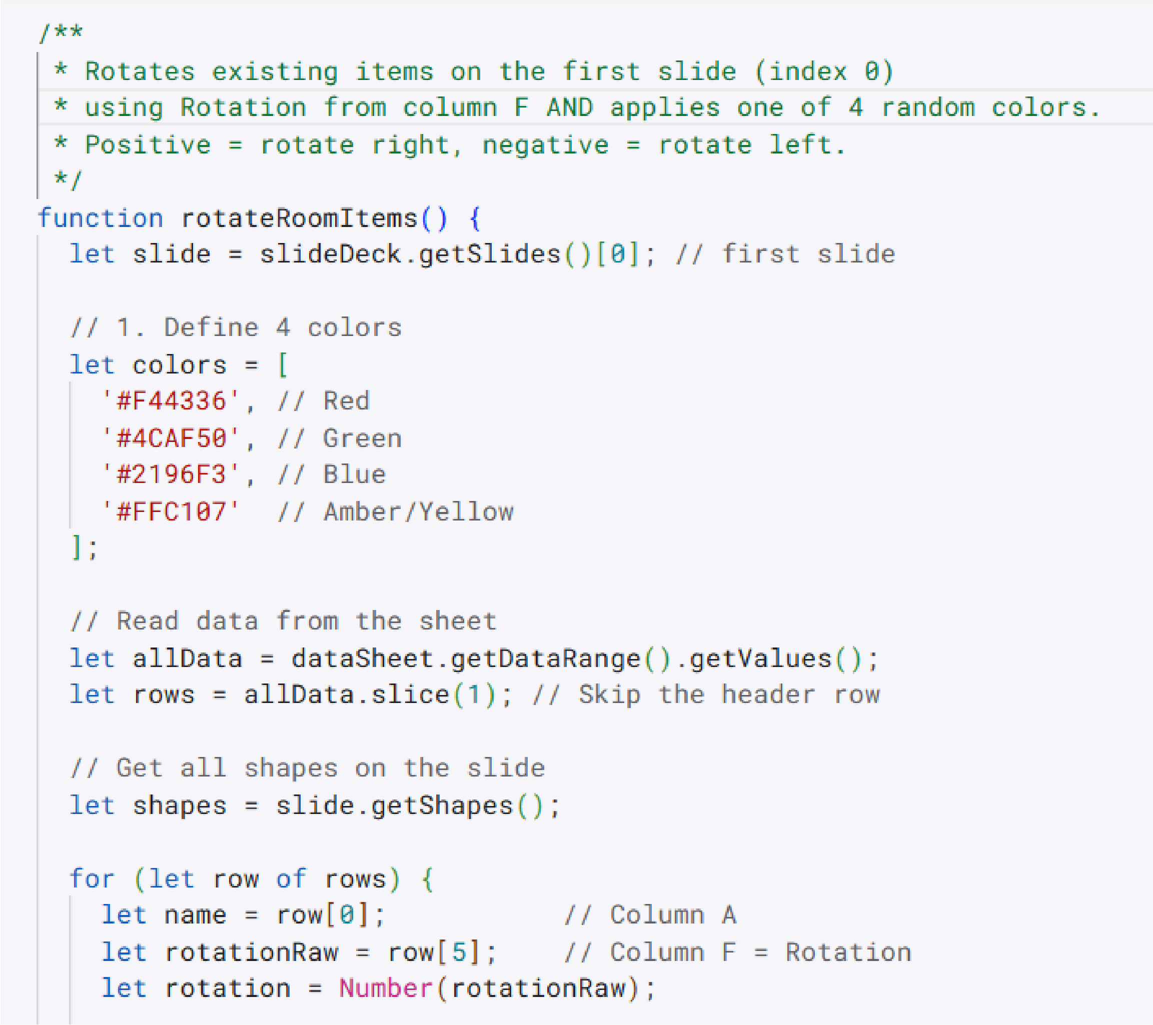
```
9:31:44 PM   Notice   Execution started
9:31:49 PM   Info     Row: name=Sofa, x=50, y=100, w=200, h=80
9:31:49 PM   Info     Drew: Sofa
9:31:49 PM   Info     Row: name=Table, x=300, y=150, w=100, h=100
9:31:49 PM   Info     Drew: Table
9:31:49 PM   Info     Row: name=TV, x=50, y=300, w=150, h=20
9:31:49 PM   Info     Drew: TV
9:31:49 PM   Info     Row: name=Shelf, x=450, y=50, w=30, h=150
9:31:49 PM   Info     Drew: Shelf
9:31:49 PM   Notice   Execution completed
```



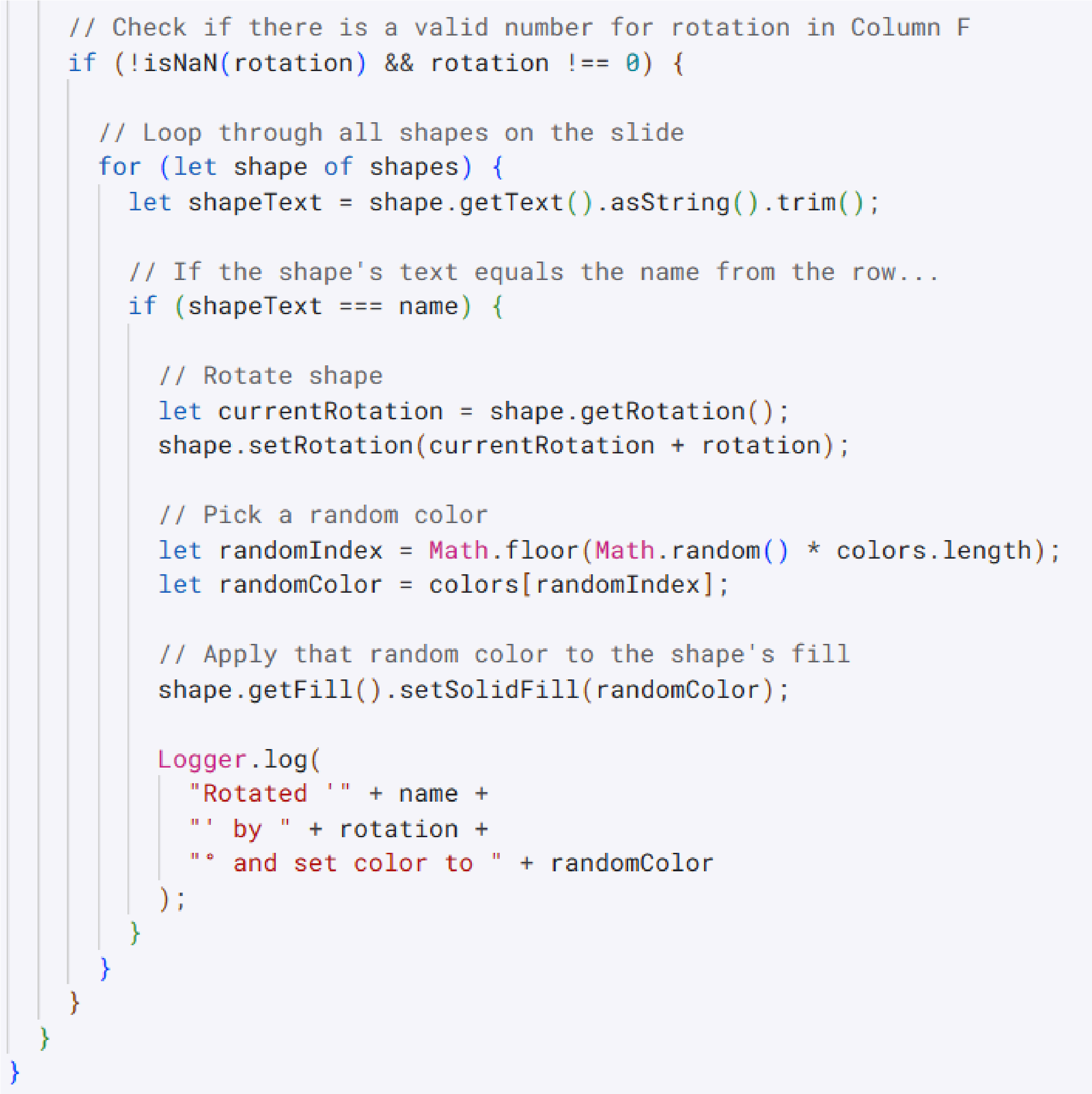
Function: `rotateRoomItems()`

This function changes the existing shapes that were drawn by the first function.
It reads rotation values from the “Rotation” column in the spreadsheet and rotates each

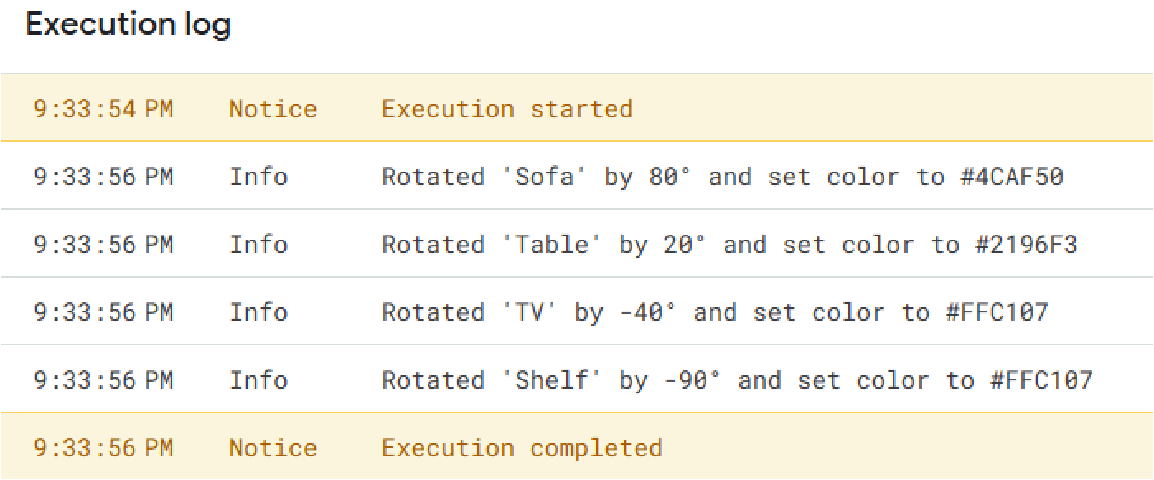
matching shape on the slide to the right or left depending on the number (positive = right, negative = left).
In addition, it gives every rotated shape one of four random colors — red, green, blue, or yellow — to make the layout more colorful and dynamic.
This shows how data in the spreadsheet can control both the position and the appearance of objects in the slide automatically.



```
/**
 * Rotates existing items on the first slide (index 0)
 * using Rotation from column F AND applies one of 4 random colors.
 * Positive = rotate right, negative = rotate left.
 */
function rotateRoomItems() {
  let slide = slideDeck.getSlides()[0]; // first slide

  // 1. Define 4 colors
  let colors = [
    '#F44336', // Red
    '#4CAF50', // Green
    '#2196F3', // Blue
    '#FFC107'  // Amber/Yellow
  ];

  // Read data from the sheet
  let allData = dataSheet.getDataRange().getValues();
  let rows = allData.slice(1); // Skip the header row

  // Get all shapes on the slide
  let shapes = slide.getShapes();

  for (let row of rows) {
    let name = row[0];           // Column A
    let rotationRaw = row[5];    // Column F = Rotation
    let rotation = Number(rotationRaw);
```

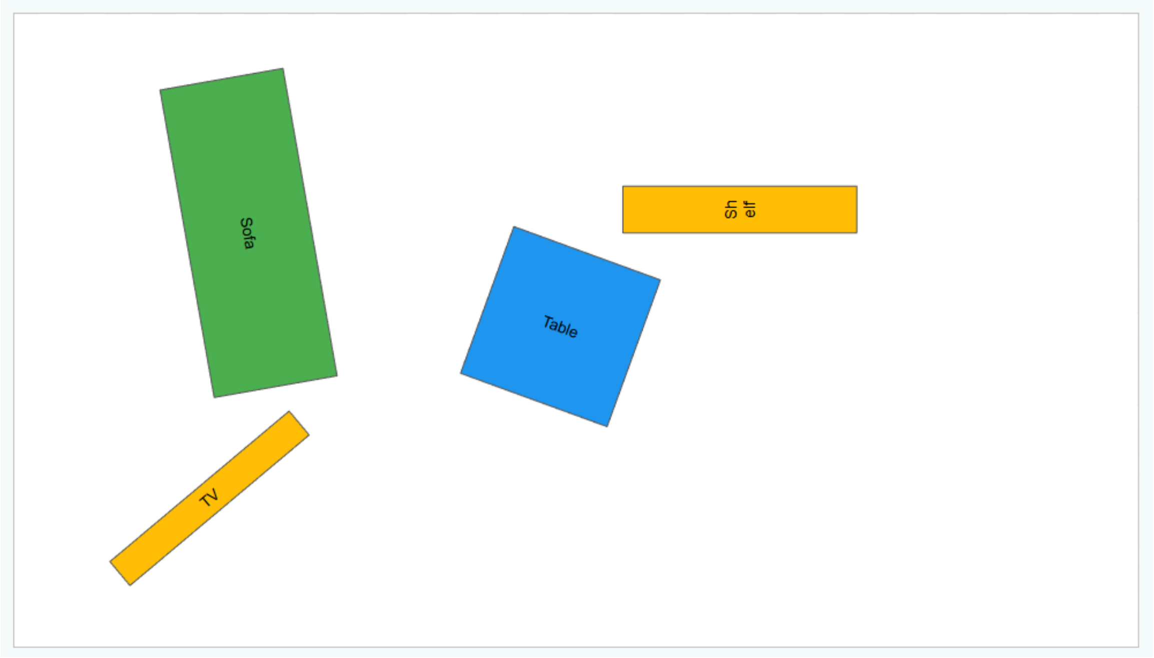


```
    // Check if there is a valid number for rotation in Column F
    if (!isNaN(rotation) && rotation !== 0) {

      // Loop through all shapes on the slide
      for (let shape of shapes) {
        let shapeText = shape.getText().asString().trim();

        // If the shape's text equals the name from the row...
        if (shapeText === name) {

          // Rotate shape
          let currentRotation = shape.getRotation();
          shape.setRotation(currentRotation + rotation);

          // Pick a random color
          let randomIndex = Math.floor(Math.random() * colors.length);
          let randomColor = colors[randomIndex];

          // Apply that random color to the shape's fill
          shape.getFill().setSolidFill(randomColor);

          Logger.log(
            "Rotated '" + name +
            "' by " + rotation +
            "° and set color to " + randomColor
          );
        }
      }
    }
  }
}
```



Execution log

```
9:33:54 PM   Notice   Execution started
9:33:56 PM   Info     Rotated 'Sofa' by 80° and set color to #4CAF50
9:33:56 PM   Info     Rotated 'Table' by 20° and set color to #2196F3
9:33:56 PM   Info     Rotated 'TV' by -40° and set color to #FFC107
9:33:56 PM   Info     Rotated 'Shelf' by -90° and set color to #FFC107
9:33:56 PM   Notice   Execution completed
```


Sofa
Sh elf
Table
TV