Homework Assignment – Updating presentation slides with Google Apps Script

Report by Riina Kikkas

Spreadsheet link

Scenario Explanation

In this project I created a small dashboard for cats stored in a Google Sheets file.

The script reads the data about each cat (name, breed, age, status) and updates a Google Slides presentation.

The presentation automatically shows the total number of cats, the number of cats per breed and individual cards for each cat.

addQuoteFromSheetToFirstSlide()

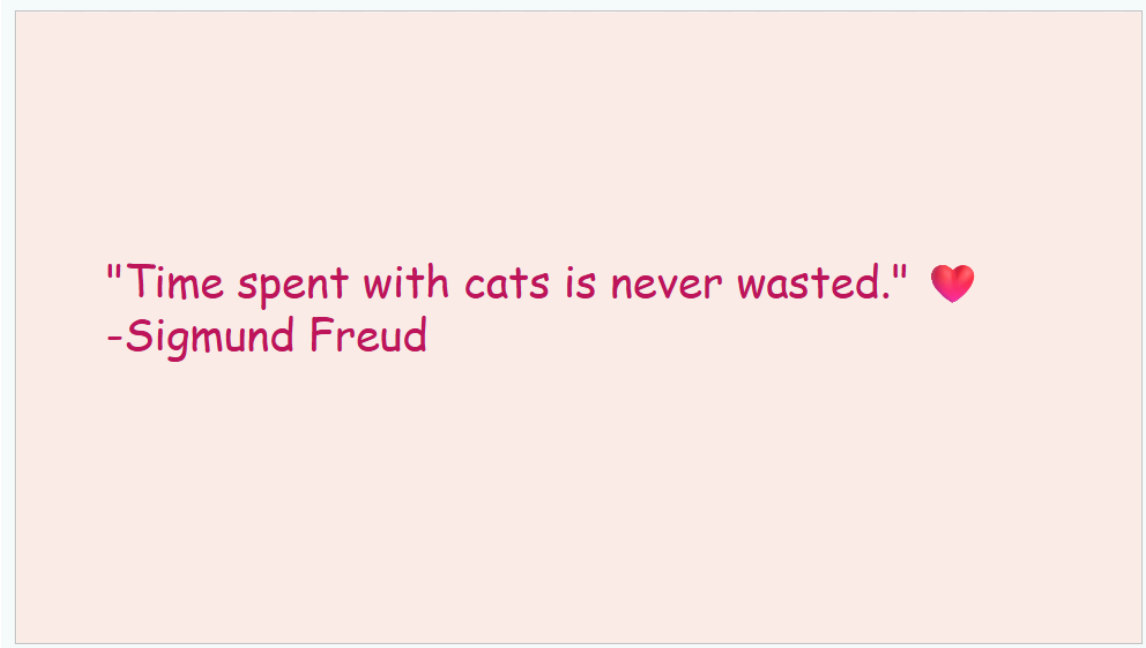
This function updates the first slide with a quote taken from the spreadsheet.

- It opens the presentation and selects slide 1 (index 0).
- It deletes all existing shapes on that slide, so the slide is clean.
- It sets a light pink background color for the whole slide.
- It opens the spreadsheet and reads two cells from the second sheet:
  - A1 → the quote text
  - A2 → the author
- It creates a new text box in the center of the slide.
- It writes the quote in the format "quote" ❤️ on the first line and – author on the second line.
- It centers the text vertically inside the box and applies styling:
  - font size 28
  - dark pink/red color
  - Comic Sans MS font

ID	Name	Breed	Age	Status
1	Luna	British Shorthair	2	Available
2	Miku	Siamese	5	Adopted
3	Bella	Maine Coon	3	Available
4	Tommi	Siamese	7	Available
5	Nurrik	Domestic Shorthair	10	Adopted

```
function addQuoteFromSheetToFirstSlide() {
  // Open the presentation and select the first slide (index 0)
  let presentation = SlidesApp.openById("1eaGIjK5vI3oZJuaaSa-OAP3OnCYHyPyGBP0Jy5gP4M0");
  let slide = presentation.getSlides()[0];

  // Clear all shapes on this slide
  for (let shape of slide.getShapes()) {
    shape.remove();
  }

  // Set a light red background for the whole slide
  slide.getBackground().setSolidFill("#fdecea"); // light red

  // Open the spreadsheet and get the second sheet (index 1)
  let ss = SpreadsheetApp.openById("1ol2mQbv_u16EyXd0GwEQ5WpfBhtXCXHLibS9hPQnobw");
  let quoteSheet = ss.getSheets()[1]; // A1 = quote, A2 = author

  // Read quote and author from the sheet
  let quote = quoteSheet.getRange("A1").getValue();
  let author = quoteSheet.getRange("A2").getValue();

  // Create a text box in the center area
  let quoteShape = slide.insertShape(
    SlidesApp.ShapeType.TEXT_BOX,
    50,
    140,
    580,
    100
  );
```

```
  // Set the text with hearts at the beginning and at the end
  let textRange = quoteShape.getText();
  textRange.setText(
    "\"" + quote + "\" ❤️\n" +
    "-" + author
  );

  // Center the text inside the box
  quoteShape.setContentAlignment(SlidesApp.ContentAlignment.MIDDLE);

  // Style the whole text: Comic Sans, dark pink/red
  let style = textRange.getTextStyle();
  style
    .setFontSize(28)
    .setForegroundColor("#c2185b")     // dark pink
    .setFontFamily("Comic Sans MS")    // Comic Sans
    .setBold(false);
}
```



showCatCount()

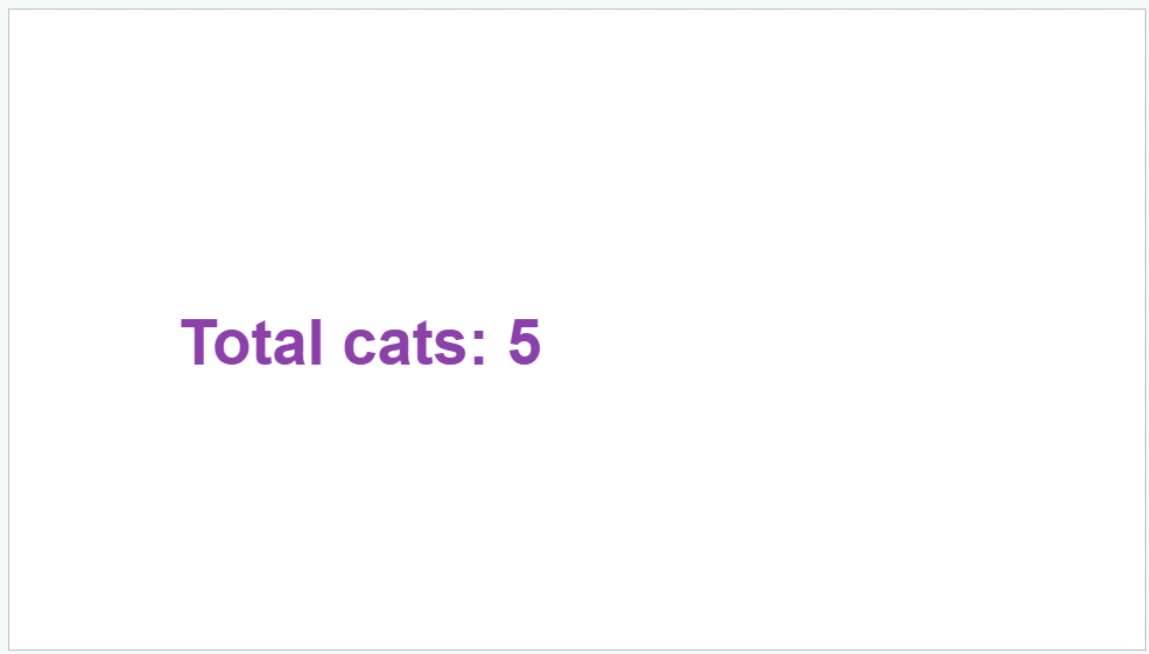
This function shows the total number of cats on the second slide.

- It opens the presentation and selects slide 2 (index 1).
- It opens the spreadsheet with cat data (first sheet) and counts how many rows there are.
- It subtracts 1 to ignore the header row, so the result is the total number of cats.
- It deletes all old shapes on the second slide.
- It creates a new text box and writes Total cats: X, where X is the calculated number.
- It formats the text to be big and visible:
    - font size 40
    - bold
    - purple text color
- It fills the text box with a white background and centers the text.

This function keeps slide 2 up to date with the current total number of cats in the spreadsheet.

```
// 2) Show the total number of cats on the second slide
function showCatCount() {
  // Open the Slides presentation and select the second slide (index 1)
  let slide = SlidesApp
    .openById("1eaGIjK5vI3oZJuaaSa-OAP3OnCYHyPyGBP0Jy5gP4M0")
    .getSlides()[1];

  // Open the spreadsheet that contains the cat data
  let sheet = SpreadsheetApp
    .openById("1ol2mQbv_u16EyXd0GwEQ5WpfBhtXCXHLibS9hPQnobw")
    .getSheets()[0];

  // Calculate the total number of cats (subtracting the header row)
  let count = sheet.getDataRange().getNumRows() - 1;

  // Remove all old shapes from this slide
  for (let shape of slide.getShapes()) {
    shape.remove();
  }

  // Create a clean new text box
  let shape = slide.insertShape(
    SlidesApp.ShapeType.TEXT_BOX,
    100,
    150,
    500,
    120
  );

  let text = shape.getText();
  text.setText("Total cats: " + count);
```

```
  let style = text.getTextStyle();
  style
    .setFontSize(40)
    .setBold(true)
    .setForegroundColor("#8e44ad");

  // Add background for clarity
  shape.getFill().setSolidFill("#ffffff");
  shape.setContentAlignment(SlidesApp.ContentAlignment.MIDDLE);
}
```



showBreedCounts()

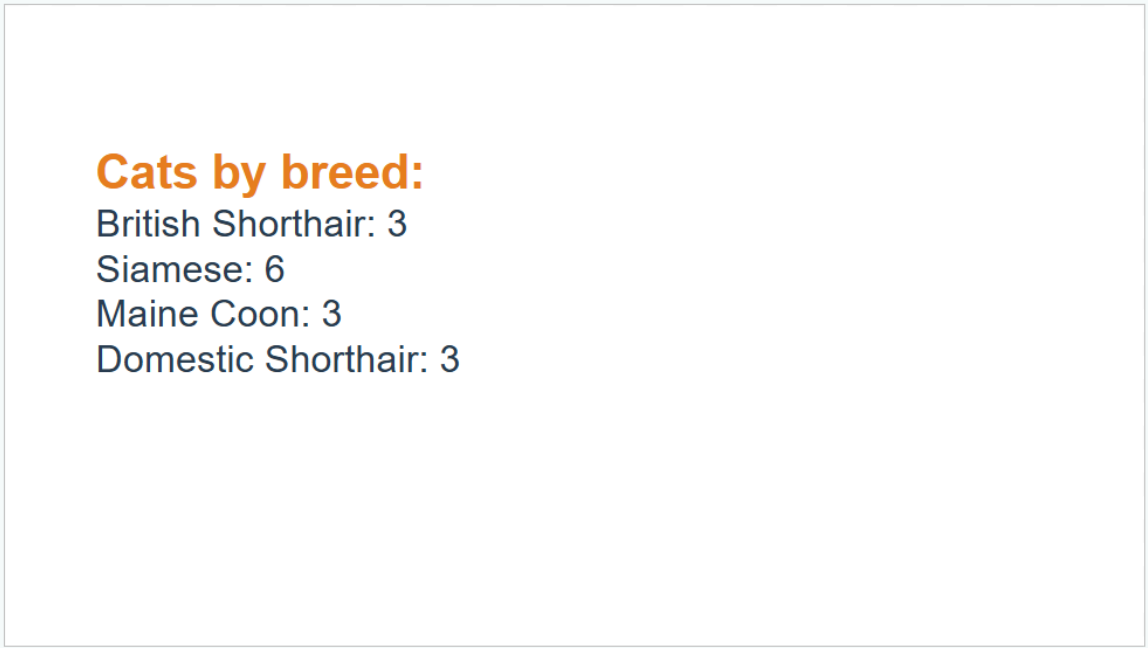
This function shows how many cats there are per breed on the third slide.

- It opens the presentation and selects slide 3 (index 2).
- It removes all existing shapes from that slide.
- It opens the cat data sheet and reads all values.
- It loops over all rows (skipping the header) and counts how many times each breed appears.
    - The breed is taken from column C (index 2).
    - The counts are stored in a JavaScript object breedCounts.
- It builds a text summary that starts with the title Cats by breed: and then adds one line per breed in the form Breed: number.
- It inserts a new text box on the slide and sets the summary text.
- It styles the text:
    - base text: font size 24, dark gray
    - the title line “Cats by breed:” is formatted separately with font size 30, bold and orange.

This function automatically generates a small statistics slide that shows the distribution of cats by breed.

```
// 3) Show the number of cats per breed on the third slide
function showBreedCounts() {
  // Open the Slides presentation and select the third slide (index 2)
  let slide = SlidesApp
    .openById("1eaGIjK5vI3oZJuaaSa-OAP3OnCYHyPyGBP0Jy5gP4M0")
    .getSlides()[2];

  // Remove all existing shapes
  for (let shape of slide.getShapes()) {
    shape.remove();
  }

  // Open the spreadsheet with cat data
  let sheet = SpreadsheetApp
    .openById("1ol2mQbv_u16EyXd0GwEQ5WpfBhtXCXHLibS9hPQnobw")
    .getSheets()[0];

  let data = sheet.getDataRange().getValues();

  // Create an object to store counts by breed
  let breedCounts = {};

  // Breed is in column C → index 2
  for (let i = 1; i < data.length; i++) {
    let breed = data[i][2];
    breedCounts[breed] = (breedCounts[breed] || 0) + 1;
  }

  // Prepare the text summary
  let text = "Cats by breed:\n";
  for (let breed in breedCounts) {
    text += `${breed}: ${breedCounts[breed]}\n`;
  }
```

```
  // Insert a new text box into the slide
  let shape = slide.insertShape(
    SlidesApp.ShapeType.TEXT_BOX,
    50,
    80,
    500,
    350
  );

  let textRange = shape.getText();
  textRange.setText(text);

  // Apply styling to the list
  let style = textRange.getTextStyle();
  style
    .setFontSize(24)
    .setBold(false)
    .setForegroundColor("#2c3e50"); // Dark gray text

  // Format the title line separately
  let titleRange = textRange.getRange(0, "Cats by breed:".length);
  titleRange.getTextStyle()
    .setFontSize(30)
    .setBold(true)
    .setForegroundColor("#e67e22"); // Orange for emphasis
}
```



highlightIfManyCats()

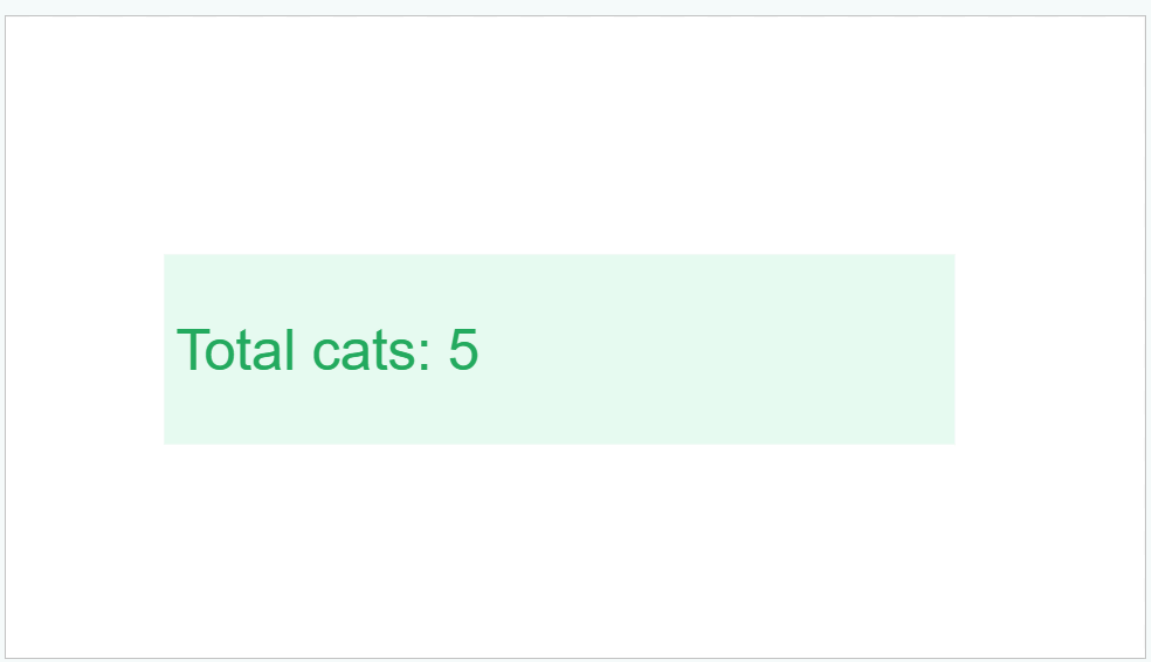
This function adds conditional formatting to the total cats slide (slide 2).

- It opens the presentation and selects slide 2 (index 1).
- It opens the cat data sheet again and counts the number of cats (rows minus header).
- It assumes that the first shape on the slide is the text box created by showCatCount().
- It updates the text to Total cats: X to make sure it is fresh.
- Then it applies different styling depending on the number of cats:
    - If there are 10 or more cats:
        - font size 48
        - bold
        - red text color
        - light red background for the text box
    - If there are fewer than 10 cats:
        - font size 36
        - not bold
        - green text color
        - light green background
- It keeps the text centered inside the box.

This function makes the slide more dynamic by visually highlighting when there are “many” cats.

```
// 5) Visual enhancement: highlight the total number of cats on the second slide
function highlightIfManyCats() {
  // Work on the second slide (index 1), same as showCatCount
  let slide = SlidesApp
    .openById("1eaGIjK5vI3oZJuaaSa-OAP3OnCYHyPyGBP0Jy5gP4M0")
    .getSlides()[1];

  // Open the spreadsheet
  let sheet = SpreadsheetApp
    .openById("1ol2mQbv_u16EyXd0GwEQ5WpfBhtXCXHLibS9hPQnobw")
    .getSheets()[0];

  // Count cats
  let count = sheet.getDataRange().getNumRows() - 1;

  // Assume first shape is the "Total cats" text box created by showCatCount
  let shape = slide.getShapes()[0];
  let text = shape.getText();

  // Update the text (in case data changed)
  text.setText("Total cats: " + count);

  // Conditional styling
  let style = text.getTextStyle();
```

```
  if (count >= 10) {
    style
      .setFontSize(48)
      .setBold(true)
      .setForegroundColor("#c0392b");
    shape.getFill().setSolidFill("#fdecea"); // light red
  } else {
    style
      .setFontSize(36)
      .setBold(false)
      .setForegroundColor("#27ae60");
    shape.getFill().setSolidFill("#eafaf1"); // light green
  }

  shape.setContentAlignment(SlidesApp.ContentAlignment.MIDDLE);
}
```



catCards()

This function generates individual cards for each cat starting from slide 4.

- It opens the presentation.
- It removes all slides after the first four, so old cards from previous runs are deleted.
- It selects slide 4 (index 3) and removes all shapes from that slide to start clean.
- It opens the cat data sheet and reads all rows.
- It defines the layout for the cards:
    - starting position on the slide (startX, startY)
    - card size (cardWidth, cardHeight)
    - gaps between columns and rows (colGap, rowGap)
    - 2 cards per row, 2 rows per slide → 4 cards per slide
- It loops over each cat (skipping the header row) and:
    - calculates which slide, row and column the card should be placed on
    - if the current slide is full, it creates a new blank slide
    - reads the cat's name, breed, age and status from the spreadsheet
    - creates a rectangular shape (the card) at the correct position
    - sets the background color of the card to light pink
    - sets a dark brown border with weight 2
    - writes the text on the card in the format:
        - first line: name
        - second line: breed • age years

        - third line: Status: status
    - applies text styling:
        - base text: font size 14, dark brown
        - the name is bold and slightly larger (font size 16)

This function builds all the detailed cat cards and places them neatly across multiple slides.

```
function catCards() {
  // Open presentation
  let presentation = SlidesApp
    .openById("1eaGIjK5vI3oZJuaaSa-OAP3OnCYHyPyGBP0Jy5gP4M0");

  // Remove previously generated slides (keep first 4)
  let slides = presentation.getSlides();
  for (let s = slides.length - 1; s >= 4; s--) {
    slides[s].remove();
  }

  // Starting slide for card generation
  let slide = presentation.getSlides()[3];

  // Clear old shapes
  for (let shape of slide.getShapes()) {
    shape.remove();
  }

  // Load cat data
  let sheet = SpreadsheetApp
    .openById("1ol2mQbv_u16EyXd0GwEQ5WpfBhtXCXHLibS9hPQnobw")
    .getSheets()[0];

  let data = sheet.getDataRange().getValues();

  // Layout settings
  let startX = 40;
  let startY = 40;

  let cardWidth = 250;
  let cardHeight = 120;
```

```
let colGap = 40;
let rowGap = 20;

let cardsPerRow = 2;
let rowsPerSlide = 2;
let cardsPerSlide = cardsPerRow * rowsPerSlide;

// Card background color
const CARD_BG_COLOR = "#fdecea";

for (let i = 1; i < data.length; i++) {
  let index = i - 1;

  // Create new slide when full
  if (index > 0 && index % cardsPerSlide === 0) {
    slide = presentation.appendSlide(SlidesApp.PredefinedLayout.BLANK);
  }

  // Card position
  let positionInSlide = index % cardsPerSlide;
  let row = Math.floor(positionInSlide / cardsPerRow);
  let col = positionInSlide % cardsPerRow;

  let x = startX + col * (cardWidth + colGap);
  let y = startY + row * (cardHeight + rowGap);

  // Read data fields
  let name = data[i][1];
  let breed = data[i][2];
  let age = data[i][3];
  let status = data[i][4];
```

```
    // Create card shape
    let card = slide.insertShape(
      SlidesApp.ShapeType.RECTANGLE,
      x,
      y,
      cardWidth,
      cardHeight
    );

    card.getFill().setSolidFill(CARD_BG_COLOR);

    // Card border
    let border = card.getBorder();
    border.setWeight(2);
    border.getLineFill().setSolidFill("#5d4037");

    // Card text
    let cardText =
      name + "\n" +
      breed + " • " + age + " years\n" +
      "Status: " + status;

    let textRange = card.getText();
    textRange.setText(cardText);

    // Base text style
    let style = textRange.getTextStyle();
    style
      .setFontSize(14)
      .setForegroundColor("#5d4037")
      .setBold(false);
```

```
      // Name styling
      textRange
        .getRange(0, name.length)
        .getTextStyle()
        .setBold(true)
        .setFontSize(16);
    }
  }
```

Luna
British Shorthair • 2 years
Status: Available

Miku
Siamese • 5 years
Status: Adopted

Bella
Maine Coon • 3 years
Status: Available

Tommi
Siamese • 7 years
Status: Available

Nurrik
Domestic Shorthair • 10 years
Status: Adopted