

# Google Apps Script for Google Slides: Comprehensive Guide



<b>What is Google Apps Script for Slides?</b>	<b>2</b>
<b>How to Use Google Apps Script with Slides</b>	<b>2</b>
<b>Key SlidesApp Methods</b>	<b>2</b>
<b>Basic Examples</b>	<b>2</b>
Example 1: Create a New Slide	2
Example 2: Add a Text Box	3
Example 3: Insert an Image	3
<b>Advanced Examples</b>	<b>3</b>
Example 4: Apply Text Styles	3
Example 5: Create a Custom Slide Deck	4
<b>Exercises</b>	<b>4</b>
Exercise 1: Add a Bullet List	4
Solution:	4
Exercise 2: Duplicate a Slide	5
Solution:	5
Exercise 3: Highlight a Specific Word	5

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

Solution:	5
<b>Multiple-Choice Questions</b>	<b>6</b>
Question 1:	6
Question 2:	6
Question 3:	6
<b>Advanced Example: Generate a Presentation with Data</b>	<b>7</b>
<b>Best Practices</b>	<b>7</b>

Google Apps Script for Google Slides allows you to automate and customize presentations. This guide covers the basics, key methods, and advanced examples, with exercises and multiple-choice questions to help you master Google Slides automation.

## What is Google Apps Script for Slides?

Google Apps Script enables you to programmatically create, modify, and manage Google Slides presentations using JavaScript. It can automate tasks like creating slides, adding content, and applying styles.

## How to Use Google Apps Script with Slides

1. Open the [Apps Script Editor](#).
2. Create a new script.
3. Save and authorize the script to access Google Slides.

## Key SlidesApp Methods

1. `SlidesApp.getActivePresentation()`: Access the active presentation.
2. `createSlide()`: Add new slides.
3. `getSlides()`: Retrieve all slides in a presentation.
4. `insertTextBox()`: Add text boxes.
5. `insertImage()`: Add images.

## Basic Examples

### Example 1: Create a New Slide

```
function createSlide() {
  const presentation = SlidesApp.getActivePresentation();
  const slide =
presentation.appendSlide(SlidesApp.PredefinedLayout.TITLE_AND_BODY);
```

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

```
slide.getPlaceholder(SlidesApp.PlaceholderType.TITLE).asShape().setText("New Slide");
```

```
slide.getPlaceholder(SlidesApp.PlaceholderType.BODY).asShape().setText("This is the body content.");  
}
```

### Explanation:

- `appendSlide(layout)`: Adds a new slide with a predefined layout.
- `getPlaceholder(type)`: Accesses placeholders (e.g., title, body).
- `setText()`: Sets text for placeholders.

### Example 2: Add a Text Box

```
function addTextBox() {  
  const slide = SlidesApp.getActivePresentation().getSlides()[0];  
  slide.insertTextBox("Hello, Apps Script!", 100, 100, 400, 50);  
}
```

### Explanation:

- `insertTextBox(text, left, top, width, height)`: Adds a text box at specified coordinates with dimensions.

### Example 3: Insert an Image

```
function addImage() {  
  const slide = SlidesApp.getActivePresentation().getSlides()[0];  
  const imageUrl = "https://via.placeholder.com/150";  
  slide.insertImage(imageUrl, 50, 50, 200, 150);  
}
```

### Explanation:

- `insertImage(url, left, top, width, height)`: Adds an image at specified coordinates with dimensions.

## Advanced Examples

### Example 4: Apply Text Styles

```
function styleText() {
```

*Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis*

```

const slide = SlidesApp.getActivePresentation().getSlides()[0];
const textBox = slide.insertTextBox("Styled Text", 100, 100, 400,
50);
const textRange = textBox.getText().getRange(0, 11);

textRange.setFontFamily("Arial").setFontSize(24).setForegroundColor("#
ff0000");
}

```

### Explanation:

- `getRange(start, end)`: Selects a range of text.
- `setFontFamily()`, `setFontSize()`, `setForegroundColor()`: Apply styles to text.

### Example 5: Create a Custom Slide Deck

```

function createSlideDeck() {
  const presentation = SlidesApp.create("My Slide Deck");
  const slide =
presentation.appendSlide(SlidesApp.PredefinedLayout.TITLE);

slide.getPlaceholder(SlidesApp.PlaceholderType.TITLE).asShape().setTex
t("Welcome to My Slide Deck");
  slide.getNotesPage().getSpeakerNotesShape().setText("Remember to
greet the audience.");
}

```

### Explanation:

- `SlidesApp.create(name)`: Creates a new presentation.
- `getNotesPage().getSpeakerNotesShape()`: Adds speaker notes to a slide.

## Exercises

### Exercise 1: Add a Bullet List

Write a script to add a slide with a bullet list containing three items: "Item 1", "Item 2", "Item 3".

### Solution:

```

function addBulletList() {

```

```

    const slide =
SlidesApp.getActivePresentation().appendSlide(SlidesApp.PredefinedLayout.TITLE_AND_BODY);

slide.getPlaceholder(SlidesApp.PlaceholderType.TITLE).asShape().setText("Bullet List");
    const body =
slide.getPlaceholder(SlidesApp.PlaceholderType.BODY).asShape();
    body.setText("Item 1\nItem 2\nItem 3");
    body.getText().getParagraphs().forEach((paragraph) =>
paragraph.setBullet(true));
}

```

### Exercise 2: Duplicate a Slide

Write a script to duplicate the first slide in the presentation.

#### Solution:

```

function duplicateSlide() {
    const presentation = SlidesApp.getActivePresentation();
    const firstSlide = presentation.getSlides()[0];
    presentation.appendSlide(firstSlide);
}

```

### Exercise 3: Highlight a Specific Word

Write a script to find and highlight the word "important" in the text on the first slide.

#### Solution:

```

function highlightWord() {
    const slide = SlidesApp.getActivePresentation().getSlides()[0];
    const textElements = slide.getShapes();
    textElements.forEach((shape) => {
        const text = shape.getText();
        if (text) {
            const indices = [];
            let index = text.asString().indexOf("important");
            while (index !== -1) {
                indices.push(index);
                index = text.asString().indexOf("important", index + 1);
            }
        }
    });
}

```

*Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis*

```
    }
    indices.forEach((start) => {
        const range = text.getRange(start, start +
"important".length);
        range.setForegroundColor("#FFFF00"); // Highlight in yellow
    });
    }
});
}
```

## Multiple-Choice Questions

### Question 1:

Which method is used to add a new slide to a presentation?

1. SlidesApp.addSlide()
2. SlidesApp.appendSlide()
3. SlidesApp.createSlide()
4. SlidesApp.newSlide()

**Answer:** 2. SlidesApp.appendSlide()

### Question 2:

What does setText("Welcome") do?

1. Adds a new slide with the text "Welcome".
2. Sets the title of the presentation to "Welcome".
3. Sets the content of a text box or placeholder to "Welcome".
4. Highlights the text "Welcome".

**Answer:** 3. Sets the content of a text box or placeholder to "Welcome".

### Question 3:

Which method is used to add an image to a slide?

1. addImage(url)
2. insertImage(url)
3. appendImage(url)
4. addPicture(url)

*Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis*

**Answer:** 2. `insertImage(url)`

## Advanced Example: Generate a Presentation with Data

```
function generatePresentation() {
  const presentation = SlidesApp.create("Sales Report");
  const data = [
    ["Q1", "$10,000"],
    ["Q2", "$15,000"],
    ["Q3", "$12,000"],
    ["Q4", "$18,000"]
  ];
  data.forEach((row) => {
    const slide =
presentation.appendSlide(SlidesApp.PredefinedLayout.TITLE_AND_BODY);

slide.getPlaceholder(SlidesApp.PlaceholderType.TITLE).asShape().setText(`Sales: ${row[0]}`);

slide.getPlaceholder(SlidesApp.PlaceholderType.BODY).asShape().setText(`Revenue: ${row[1]}`);
  });
}
```

### Explanation:

- Iterates over data to create slides dynamically.
- Adds quarterly sales data to each slide.

## Best Practices

1. **Use placeholders:** Utilize predefined layouts for structured slides.
2. **Optimize scripts:** Avoid unnecessary API calls to reduce execution time.
3. **Test thoroughly:** Verify scripts on sample presentations before deploying them.