



 *Supercharge Your Google Workspace:*

Google Apps Script

Automated Attendance Tracker and Email

Reminder  

Exercise: Automated Attendance Tracker and Email Reminder	2
Objective:	2
Google Sheet Setup:	3
Script Functionality:	3
Email Content:	4
Triggers:	4
Deliverables:	4
Hints:	4
Bonus Challenge:	4
Step 1: Setup Google Sheet	5
Step 2: Writing the Google Apps Script	5
Part 1: Fetch Data from Sheets	5
Part 2: Send Email Reminders	7
Step 3: Setting Up a Time-Driven Trigger	8

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

A	B	C
Date	Student Name	Yes/No
Jan 17 2022	Laurence	yes
Jan 18 2022	Laurence	no
Jan 19 2022	Laurence	no
Jan 20 2022	Laurence	no
Jan 21 2022	Laurence	no
Jan 22 2022	Laurence	yes
Jan 23 2022	Laurence	yes

Exercise: Automated Attendance Tracker and Email Reminder

Objective:

Create a standalone Google Apps Script that automates the process of tracking attendance in a Google Sheet and sending email reminders to those who have been absent for a certain number of days.

Requirements:

Date	Student Name	Yes/No
Jan 17 2022	Laurence	yes
Jan 18 2022	Laurence	no

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

Jan 19 2022	Laurence	no
Jan 20 2022	Laurence	no
Jan 21 2022	Laurence	no
Jan 22 2022	Laurence	yes
Jan 23 2022	Laurence	yes

Student Name	Email	Absences
Laurence	***es@gmail.com	

Google Sheet Setup:

- Create a Google Sheet with two sheets: "Attendance" and "Students".
- The "Attendance" sheet should have columns for "Date", "Student Name", and "Present" (Yes/No).
- The "Students" sheet should have columns for "Student Name", "Email", and "Absences".

Script Functionality:

- The script should read the attendance data from the "Attendance" sheet.
- For each student, calculate the number of absences.
- Update the "Absences" column in the "Students" sheet with this number.
- If a student has been absent for 3 or more days, send an automated email reminder to the student's email address.

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

Email Content:

- The email should include the student's name, the number of absences, and a reminder message.

Triggers:

- Set up a time-driven trigger that runs the script daily.

Deliverables:

- The script code.
- A sample Google Sheet set up as per the requirements.
- Instructions on how to set up the time-driven trigger.

Hints:

- Use the SpreadsheetApp service to interact with Google Sheets.
- Use the MailApp service to send emails.
- Remember to check for empty rows and handle them appropriately to avoid errors.

Bonus Challenge:

- Enhance the script to handle weekends and holidays (assume a predefined list of holidays).
- Add a feature to send a summary email to the teacher with the names of students who have been absent for 3 or more days.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

This exercise will help you develop skills in reading and writing data to Google Sheets, sending emails, and setting up triggers in Google Apps Script. It's a practical scenario that combines various aspects of automation using Google's platform.

Step 1: Setup Google Sheet

First, you need to create a Google Sheet with the following structure:

- Sheet 1 ("Attendance"):
 - Columns: "Date", "Student Name", "Present" (Yes/No)
- Sheet 2 ("Students"):
 - Columns: "Student Name", "Email", "Absences"

Step 2: Writing the Google Apps Script

We'll write a standalone script. Go to [Google Apps Script](#) and start a new project.

Part 1: Fetch Data from Sheets

```
function updateAttendance() {  
  const ss = SpreadsheetApp.getActiveSpreadsheet();  
  const attendanceSheet = ss.getSheetByName("Attendance");  
  const studentSheet = ss.getSheetByName("Students");  
  
  const attendanceData = attendanceSheet.getDataRange().getValues();
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```

const studentData = studentSheet.getDataRange().getValues();

const studentAbsenceMap = {};

// Processing attendance data
attendanceData.shift(); // Remove header row
for (let i = 0; i < attendanceData.length; i++) {
  const row = attendanceData[i];
  const studentName = row[1];
  const present = row[2].toLowerCase() === 'yes';

  if (!studentAbsenceMap[studentName]) {
    studentAbsenceMap[studentName] = 0;
  }

  studentAbsenceMap[studentName] = present ? 0 :
  studentAbsenceMap[studentName] + 1;
}

// Updating student sheet with absences
for (let j = 1; j < studentData.length; j++) { // Skip header
  const studentName = studentData[j][0];
  if (studentAbsenceMap[studentName] !== undefined) {
    studentSheet.getRange(j + 1, 3).setValue(studentAbsenceMap[studentName]);
  }
}

```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
 Courses <https://basescripts.com/>

```
}  
}  
}
```


Part 2: Send Email Reminders

```
function sendEmailReminders() {  
  const studentSheet =  
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Students");  
  const studentData = studentSheet.getDataRange().getValues();  
  
  studentData.shift(); // Remove header row  
  for (let i = 0; i < studentData.length; i++) {  
    const row = studentData[i];  
    const studentName = row[0];  
    const email = row[1];  
    const sAbsences = row[2];  
  
    if (sAbsences >= 3) {  
      const subject = 'Attendance Reminder';  
      const message = 'Dear ' + studentName + ',\n\n' +  
      'You have been absent for ' + sAbsences + ' consecutive days. ' +  
      'Please ensure to attend the upcoming classes.\n\n' +  
      'Best,\n' +
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
'Your Teacher';  
MailApp.sendEmail(email, subject, message);  
}  
}  
}
```

Step 3: Setting Up a Time-Driven Trigger

- In the Google Apps Script editor, click on the clock icon () on the left side to open "Current project's triggers".
- Click on "+ Add Trigger" in the bottom right corner.
- Select the updateAttendance function.
- Choose the type of trigger (e.g., "Day timer" for daily execution).
- Select the time of the day when you want the script to run.

Explanation

- Fetching Data: The script reads data from both "Attendance" and "Students" sheets. It calculates the absences for each student.
- Updating Sheet: It then updates the "Absences" column in the "Students" sheet with this calculated value.
- Sending Emails: If a student's absences are 3 or more, an email is sent using the MailApp service.
- Time-Driven Trigger: This setup allows the script to run automatically at a specified time every day.

Final Steps

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

- Test the script manually by running `updateAttendance` and `sendEmailReminders` from the script editor.
- Ensure you have provided the necessary permissions for the script to read from the Google Sheet and send emails.

This script provides a basic yet effective way to automate attendance tracking and sending reminders, demonstrating the power of Google Apps Script in automating tasks involving Google Workspace tools.

```
function updateAttendance() {  
  const ss = SpreadsheetApp.getActiveSpreadsheet();  
  const attendanceSheet = ss.getSheetByName("Attendance");  
  const studentSheet = ss.getSheetByName("Students");  
  
  const attendanceDataAll = attendanceSheet.getDataRange().getValues();  
  const studentData = studentSheet.getDataRange().getValues();  
  const attendanceData = attendanceDataAll.slice(1);  
  const studentAbsenceMap = {};  
  for (let i = 0; i < attendanceData.length; i++) {  
    const row = attendanceData[i];  
    const studentName = row[1];  
    const present = row[2].toLowerCase() === 'yes';  
    if (!studentAbsenceMap[studentName]) {  
      studentAbsenceMap[studentName] = 0;  
    }  
  }  
}
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```

    studentAbsenceMap[studentName] = present ?
studentAbsenceMap[studentName] : studentAbsenceMap[studentName] + 1;
}

// Updating student sheet with absences
for (let j = 1; j < studentData.length; j++) { // Skip header
    const studentName = studentData[j][0];
    if (studentAbsenceMap[studentName] !== undefined) {
        studentSheet.getRange(j + 1, 3).setValue(studentAbsenceMap[studentName]);
    }
}
}

function sendEmailReminders() {
    const studentSheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Students");
    const studentData = studentSheet.getDataRange().getValues();

    studentData.shift(); // Remove header row
    for (let i = 0; i < studentData.length; i++) {
        const row = studentData[i];
        const studentName = row[0];
        const email = row[1];

```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
 Courses <https://basescripts.com/>

```

const sAbsences = row[2];

if (sAbsences >= 3) {
  const subject = 'Attendance Reminder';
  const message = 'Dear ' + studentName + ',\n\n' +
    'You have been absent for ' + sAbsences + ' days. ' +
    'Please ensure to attend the upcoming classes.\n\n' +
    'Best,\n' +
    'Your Teacher';
  MailApp.sendEmail(email, subject, message);
}
}
}

```

```

const studentAbsenceMap = {};
for (let i = 0; i <
attendanceData.length; i++) {
  const row = attendanceData[i];
  const studentName = row[1];

```

```
    const present = row[2].toLowerCase()
=== 'yes' ;

    if (!studentAbsenceMap[studentName]) {
        studentAbsenceMap[studentName] = 0;
    }

    studentAbsenceMap[studentName] =
present ? studentAbsenceMap[studentName] :
studentAbsenceMap[studentName] + 1;
}

Logger.log(studentAbsenceMap);
```