

Apps Script **Code Exercises**



Enhance Your
Google Calendar Skills
with Apps Script!



Enhance Your Google Calendar Skills with Apps Script!




	1
Exercise 1: Create a Calendar Event	2
Exercise 2: Fetch Calendar Events	3
Exercise 3: Update an Event	3
Exercise 4: Delete a Calendar Event	4
Exercise 5: Create Recurring Events	5
Exercise 6: Send Invitations for an Event	5
Exercise 7: Working with Event Reminders	6
Exercise 8: Change Event Visibility	7
Exercise 9: Working with Multiple Calendars	7
Exercise 10: Create a Calendar Event with Advanced Options	8





Enhance Your Google Calendar Skills with Apps Script!



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

 Excited to unveil 10 dynamic coding exercises for mastering Google Apps Script's Calendar Service (CalendarApp)! From creating simple events to managing complex calendars, these exercises cover a wide range of functionalities.

 Whether you're automating meeting schedules, setting up reminders, or handling multiple calendars, these exercises are designed to elevate your proficiency and streamline your calendar management.

 Embrace the power of automation and transform your Google Calendar experience!

Exercise 1: Create a Calendar Event

Objective: Learn to create a basic event in a Google Calendar.

Explanation: This exercise introduces creating a new event in a user's default calendar.

Code:

```
function createEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var startTime = new Date('March 15, 2024 10:00:00');  
  var endTime = new Date('March 15, 2024 11:00:00');
```

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

```
var event = calendar.createEvent('Meeting', startTime, endTime);
Logger.log('Event ID: ' + event.getId());
}
```

Exercise 2: Fetch Calendar Events

Objective: Learn to retrieve events from a specific date range.

Explanation: This exercise teaches how to fetch calendar events within a given date range.

Code:

```
function fetchEvents() {
  var calendar = CalendarApp.getDefaultCalendar();
  var startDate = new Date('March 1, 2024');
  var endDate = new Date('March 31, 2024');
  var events = calendar.getEvents(startDate, endDate);
  for (var i = 0; i < events.length; i++) {
    Logger.log('Event: ' + events[i].getTitle());
  }
}
```

Exercise 3: Update an Event

Objective: Understand how to update an existing calendar event.

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

Explanation: This exercise focuses on modifying the details of an existing event.

Code:

```
function updateEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEventsForDay(new Date('March 15, 2024'));  
  if (events.length > 0) {  
    var event = events[0];  
    event.setTitle('Updated Meeting Title');  
  }  
}
```

Exercise 4: Delete a Calendar Event

Objective: Learn to delete a calendar event.

Explanation: This exercise shows how to programmatically remove an event from a calendar.

Code:

```
function deleteEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEventsForDay(new Date('March 15, 2024'));  
  if (events.length > 0) {  
    var event = events[0];  
  }  
}
```

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

```
    event.deleteEvent();  
  }  
}
```

Exercise 5: Create Recurring Events

Objective: Understand how to set up recurring events.

Explanation: This exercise demonstrates creating events that recur on a specific schedule.

Code:

```
function createRecurringEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var startTime = new Date('April 1, 2024 08:00:00');  
  var endTime = new Date('April 1, 2024 09:00:00');  
  var recurrence = CalendarApp.newRecurrence().addWeeklyRule().until(new  
Date('June 1, 2024'));  
  var event = calendar.createEventSeries('Weekly Meeting', startTime, endTime,  
recurrence);  
  Logger.log('Event Series ID: ' + event.getId());  
}
```

Exercise 6: Send Invitations for an Event

Objective: Learn to invite guests to an event.

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

Explanation: This exercise focuses on adding guests to an event and sending them invitations.

Code:

```
function inviteGuests() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var startTime = new Date('March 20, 2024 15:00:00');  
  var endTime = new Date('March 20, 2024 16:00:00');  
  var event = calendar.createEvent('Team Meeting', startTime, endTime);  
  event.addGuest('example@email.com');  
}
```

Exercise 7: Working with Event Reminders

Objective: Understand how to manage event reminders.

Explanation: This exercise shows how to add, update, and remove reminders for an event.

Code:

```
function manageReminders() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var event = calendar.getEventsForDay(new Date())[0];  
  event.removeAllReminders();  
}
```

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

```
event.addPopupReminder(30); // 30 minutes before
}
```

Exercise 8: Change Event Visibility

Objective: Learn to change the visibility of a calendar event.

Explanation: This exercise demonstrates how to set an event's visibility (public, private).

Code:

```
function changeEventVisibility() {
  var calendar = CalendarApp.getDefaultCalendar();
  var event = calendar.getEventsForDay(new Date())[0];
  event.setVisibility(CalendarApp.Visibility.PRIVATE);
}
```

Exercise 9: Working with Multiple Calendars

Objective: Learn to interact with multiple calendars in a Google account.

Explanation: This exercise focuses on retrieving and working with different calendars within a user's Google account.

Code:

```
function listCalendars() {
```

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

```
var calendars = CalendarApp.getAllCalendars();
for (var i = 0; i < calendars.length; i++) {
  Logger.log('Calendar: ' + calendars[i].getName());
}
}
```

Exercise 10: Create a Calendar Event with Advanced Options

Objective: Understand how to use advanced options when creating an event.

Explanation: This exercise teaches how to create an event with additional details like location and description.

Code:

```
function createAdvancedEvent() {
  var calendar = CalendarApp.getDefaultCalendar();
  var startTime = new Date('April 5, 2024 10:00:00');
  var endTime = new Date('April 5, 2024 11:00:00');
  var options = {
    location: 'Conference Room',
    description: 'Discuss project milestones'
  };
  var event = calendar.createEvent('Project Meeting', startTime, endTime, options);
  Logger.log('Event ID: ' + event.getId());
}
```

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

}

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