





Apps Script Coding Questions

 Exciting Insights into Google Apps Script:
Mastering CalendarApp! 

| | |
|---|----------|
| Question 1: Creating an Event | 1 |
| Question 2: Finding Events | 2 |
| Question 3: Deleting an Event | 3 |
| Question 4: Updating an Event | 3 |
| Question 5: Creating a Recurring Event | 4 |
| Question 6: Getting Event Details | 5 |
| Question 7: Adding Guests to an Event | 6 |
| Question 8: Changing Event Visibility | 6 |
| Question 9: Getting Calendar ID | 7 |
| Question 10: Listing All Calendars | 8 |

Question 1: Creating an Event

Question: How do you create a simple event in the user's default calendar using Google Apps Script?

Answer:

```
function createEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();
```

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

```
var startTime = new Date('March 15, 2024 15:00:00');  
var endTime = new Date('March 15, 2024 16:00:00');  
var event = calendar.createEvent('Meeting with Bob', startTime, endTime);  
}
```

Explanation: This script uses `CalendarApp.getDefaultCalendar()` to get the user's default calendar. It then defines start and end times for the event and creates a new event titled 'Meeting with Bob'.

Question 2: Finding Events

Question: How can you search for events with a specific title in a calendar?

Answer:

```
function findEvents() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEvents(new Date('March 1, 2024'), new  
Date('March 31, 2024'), {search: 'Meeting'});  
  return events;  
}
```

Explanation: This script searches the default calendar for events with the word 'Meeting' in their title, between March 1, 2024, and March 31, 2024. The `getEvents()` method is used with a date range and a search term.

Question 3: Deleting an Event

Question: How do you delete an event by its title?

Answer:

```
function deleteEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEvents(new Date('March 1, 2024'), new  
Date('March 31, 2024'), {search: 'Unwanted Event'});  
  if (events.length > 0) {  
    events[0].deleteEvent();  
  }  
}
```

Explanation: This code finds events titled 'Unwanted Event' and deletes the first occurrence. It's important to handle the case where no events are found to avoid errors.

Question 4: Updating an Event

Question: How do you change the time of an existing event?

Answer:

```
function updateTime() {  
  var calendar = CalendarApp.getDefaultCalendar();
```

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

```
var events = calendar.getEvents(new Date('March 15, 2024'), new
Date('March 16, 2024'), {search: 'Meeting with Bob'});
if (events.length > 0) {
var newStartTime = new Date('March 15, 2024 17:00:00');
var newEndTime = new Date('March 15, 2024 18:00:00');
events[0].setTime(newStartTime, newEndTime);
}
}
```

Explanation: This function looks for events titled 'Meeting with Bob' and updates the start and end times of the first matching event. It's crucial to check that the event exists before attempting to modify it.

Question 5: Creating a Recurring Event

Question: How do you create a weekly recurring event?

Answer:

```
function createRecurringEvent() {
var calendar = CalendarApp.getDefaultCalendar();
var startTime = new Date('March 15, 2024 09:00:00');
var endTime = new Date('March 15, 2024 10:00:00');
var recurrence =
CalendarApp.newRecurrence().addWeeklyRule().until(new Date('June 15,
2024'));
```

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

```
calendar.createEventSeries('Weekly Team Meeting', startTime, endTime,  
recurrence);  
}
```

Explanation: This script sets up a weekly recurring event named 'Weekly Team Meeting', starting on March 15, 2024, and ending on June 15, 2024. The newRecurrence() method is used to define the recurrence pattern.

Question 6: Getting Event Details

Question: How can you retrieve and log the details of all events on a specific day?

Answer:

```
function logEventDetails() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEventsForDay(new Date('March 15, 2024'));  
  for (var i = 0; i < events.length; i++) {  
    Logger.log('Event: ' + events[i].getTitle() + ', Time: ' +  
events[i].getStartTime());  
  }  
}
```

Explanation: This function retrieves all events for March 15, 2024, and logs their title and start time. The `getEventsForDay()` method is used to get the day's events.

Question 7: Adding Guests to an Event

Question: How do you add guests to an existing event?

Answer:

```
function addGuestsToEvent() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEvents(new Date('March 15, 2024'), new  
Date('March 16, 2024'), {search: 'Meeting with Bob'});  
  if (events.length > 0) {  
    events[0].addGuest('example@example.com');  
  }  
}
```

Explanation: This script finds an event named 'Meeting with Bob' and adds a guest with the email 'example@example.com'. The `addGuest()` method is used to add guests to the event.

Question 8: Changing Event Visibility

Question: How can you change an event's visibility to private?

Answer:

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

```
function makeEventPrivate() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  var events = calendar.getEvents(new Date('March 15, 2024'), new  
Date('March 16, 2024'), {search: 'Meeting with Bob'});  
  if (events.length > 0) {  
    events[0].setVisibility(CalendarApp.Visibility.PRIVATE);  
  }  
}
```

Explanation: This function changes the visibility of the first 'Meeting with Bob' event to private. The `setVisibility()` method is employed to modify the event's visibility setting.

Question 9: Getting Calendar ID

Question: How do you find and log the ID of the user's default calendar?

Answer:

```
function logCalendarId() {  
  var calendar = CalendarApp.getDefaultCalendar();  
  Logger.log(calendar.getId());  
}
```

Explanation: This simple script retrieves the ID of the user's default calendar using `getId()` and logs it. The calendar ID is often used in more advanced scripting scenarios.

Question 10: Listing All Calendars

Question: How can you list all calendar names and their IDs that the user has access to?

Answer:

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

Explanation: This script uses `getAllCalendars()` to retrieve all calendars the user has access to and logs their names and IDs. This can be useful for identifying specific calendars for more complex operations.