

## 目次

- GASとは
  - 【GASの概要】
  - 【GASで利用されるGoogleの主なサービス】
  - 【GASの2種類の方式(コンテナバインドとスタンドアロン)】
- スクリプトエディタとは
  - 【スクリプトエディタの概要】
  - 【スクリプトエディタの開き方】
  - 【スクリプトエディタの使用手順】
- トリガーとは
  - 【トリガーの概要】
  - 【トリガーの追加方法】
- ライブラリとは
  - 【ライブラリの概要】
  - 【ライブラリの実用方法】
- 公式リファレンスとは
  - 【公式リファレンスの概要】
  - 【公式リファレンスの活用方法】

## ■ GASとは

### 【GASの概要】

GASはGoogle Apps Scriptの略称で、Google が提供するGoogleのアプリケーションを簡単に操作するためのプログラミング言語です。GASはJavaScriptというWebブラウザ上で動作するプログラミング言語をベースに作成されています。

### 【GASで利用されるGoogleの主なサービス】

スプレッドシート、Gmail、ドキュメント、カレンダー、スライド、マップ、フォーム、翻訳、etc

### 【GASの2種類の方式(コンテナバインドとスタンドアロン)】

GASには大きく分けて、コンテナバインド型とスタンドアロン型の2種類の方式があります。

コンテナバインド型は、スプレッドシートやドキュメント、フォームなどのGoogleのサービスに紐づく、GASのスクリプトです。一方、スタンドアロン型は、Googleのサービスに紐づかないスクリプトです。スクリプト内でスプレッドシートやフォームなどの呼び出しは可能です。

コンテナバインド型とスタンドアロン型の特徴を表にまとめてみました。

コンテナバインド型	スタンドアロン型
<ul style="list-style-type: none"><li>・Googleの各種サービスのメニューから手軽に始められる</li><li>・紐づくGoogleサービスをID指定不要で参照できるコマンドが用意されている</li></ul>	<ul style="list-style-type: none"><li>・Googleドライブ上で表示されるため、管理しやすい</li><li>・スプレッドシートなどに紐づかないため、スプレッドシート利用者に対し、スクリプトを秘匿することができる</li><li>・設定できるトリガー* に制限がある</li></ul>

\*トリガーの詳細は「トリガーとは」を参照ください。

## ■ スクリプトエディタとは

### 【スクリプトエディタの概要】

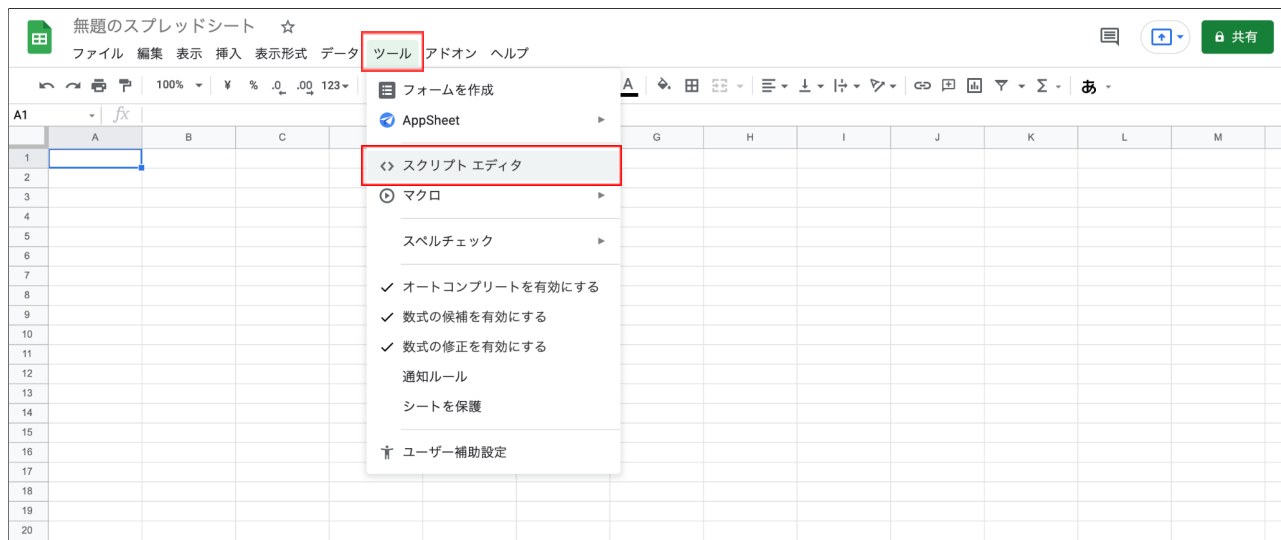
スクリプトエディタとは、スクリプトを記述し、保存することができるツールです。GASにはGoogleが用意してくれているスクリプトエディタがあり、Googleアカウントさえあれば、無料でブラウザから利用することができます。

### 【スクリプトエディタの開き方】

スクリプトエディタの開き方には2通りの方法があります。

#### <特定のGoogleサービスから開く方法>

スプレッドシートやドキュメントなどのGoogleのサービスから開く場合は、そのファイルの「ツール」タブから「スクリプトエディタ」を開きます。



「ツール」タブから開くことができるのは、スプレッドシート、ドキュメント、スライドの3サービスです。Googleフォームは、送信ボタン横のメニューから「スクリプトエディタ」を開くことができます。

また、その他のGmailやマップなどのサービスからは直接「スクリプトエディタ」を開くことはできません。

### <Googleドライブから開く方法>

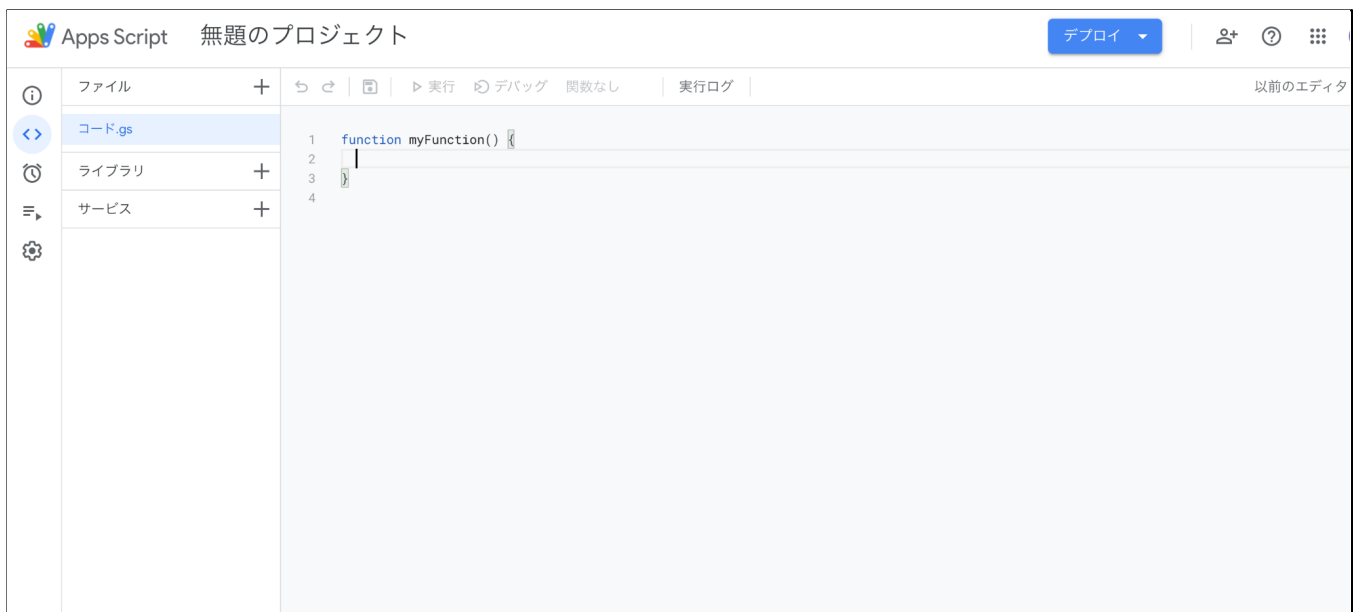
一方、特定のGoogleサービスに紐づけけない場合は、Googleドライブの「新規」から「Google Apps Script」を開きます。



どの開き方をした場合も初期のスクリプトエディタ画面は一緒です。

デフォルトでは「myFunction」という名前の関数が記載されていますが、中身は空です。

波括弧内{}にコードを記載し、実行をすると記載したコードの内容に沿った処理をしてくれます。



## 【スクリプトエディタの使用手順】

1.コードを記載し、保存します。

例として、ログを表示するスクリプトであるconsole.logに「Hello, GAS」と記載しています。

「コード.gs」の左横に表示されたオレンジ色のマークは未保存を示すマークです。

GASでは、コードは自動保存されないの、都度保存をする必要があります。

「保存」のショートカットキー:Windowsの場合「Ctrl + s」、Macの場合「Command + s」

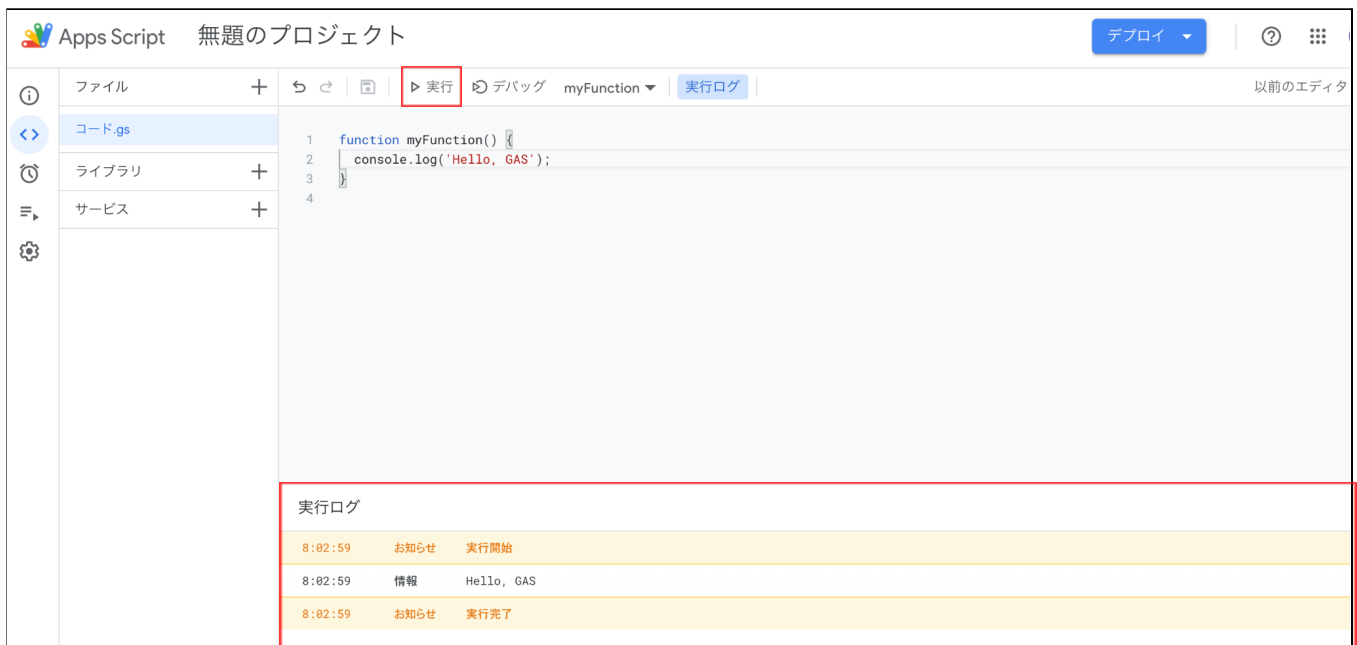


2.処理を実行します。

処理が実行されると、以下のように「実行ログ」が画面下部に表示されます。

今回の例では、ログを表示するスクリプトを記載しているので、実行ログ内に「Hello, GAS」が表示されます。

「実行」のショートカットキー:Windowsの場合「Ctrl + r」、Macの場合「Command + r」



## <発展>

過去に実行したログは、スクリプトエディタの左のメニューバーの下から2つ目のマークを選択すると確認ができます。問題なく実行できた場合はステータスが「完了」、エラーが発生した場合は「失敗しました」と表示されます。

実行数 <span>過去 7 日間の 9 個実行 を表示中</span>						
+ フィルタを追加						
導入	関数	種類	開始時間	期間	ステータス	
Head	addItem	エディタ	2021/11/02 7:16:17	4.315 秒	完了	
Head	makeInvitation	エディタ	2021/11/02 7:15:50	1.499 秒	失敗しました	
Head	copyEntryList	エディタ	2021/11/02 7:15:11	3.037 秒	失敗しました	
Head	addItem	エディタ	2021/11/01 7:45:17	3.824 秒	完了	
Head	createForm	エディタ	2021/11/01 7:36:56	8.106 秒	完了	
Head	createForm	エディタ	2021/11/01 7:33:48	8.67 秒	完了	
Head	createForm	エディタ	2021/11/01 7:32:27	14.235 秒	完了	
Head	getAccess	エディタ	2021/10/28 8:09:20	2.957 秒	完了	
Head	getAccess	エディタ	2021/10/28 8:07:53	3.331 秒	完了	

ログをクリックすると、詳細を確認することができます。

## ■ トリガーとは

### 【トリガーの概要】

GASでは指定した日時、スプレッドシートの編集、フォームの送信といったイベントに合わせて、スクリプトを実行することができる、トリガーという機能があります。

トリガーの例:「毎朝9時に〇〇を実行」「フォームが送信されたときに〇〇を実行」

### 【トリガーの追加方法】

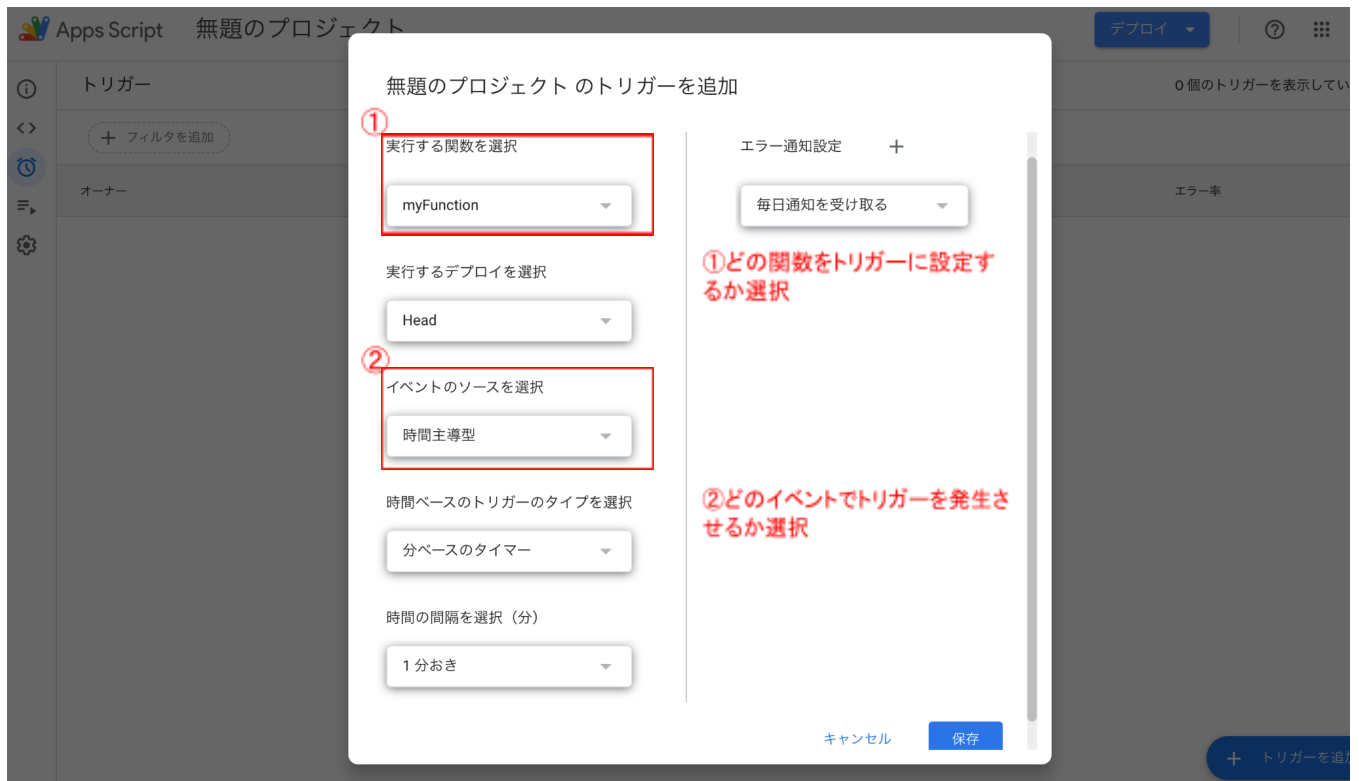
1.スクリプトエディタの左のメニューバーの時計マークを選択し、以下の画面の「新しいトリガーを作成します」もしくは「トリガーを追加」を選択します。



## 【テックジム】GASコース 補助教材

2.トリガーの内容を設定し、保存します。

例えば、先ほど「Hello, GAS」とログを表示したmyFunction関数を、1分毎に実行というトリガーを設定する場合は、以下のように設定して、保存します。



3.保存したトリガーは、以下のように一覧画面に追加されるので、設定内容に誤りがないか確認します。





4. スクリプトエディタの左のメニューバーの下から2つ目のマークを選択し、設定したトリガーが問題なく機能して、関数が実行されているか確認します。

Apps Script
無題のプロジェクト

デプロイ

?

実行数

過去 7 日間の 25 個実行 を表示中 | リアルタイムで表示

+ フィルタを追加

導入

関数

種類

開始時間

期間

ステータス

Head	myFunction	時間主導型	2021/10/24 7:24:44	0.803 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:23:44	0.193 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:22:44	0.14 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:21:44	0.23 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:20:44	0.226 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:19:44	0.22 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:18:44	0.178 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:17:44	0.369 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:16:44	0.193 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:15:44	0.107 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:14:44	0.174 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:13:44	0.273 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:12:44	0.171 秒	完了
Head	myFunction	時間主導型	2021/10/24 7:11:44	0.195 秒	完了

## ■ ライブラリとは

### 【ライブラリの概要】

GASでは、作成した関数を他のプロジェクトから利用できる「ライブラリ」という仕組みが用意されています。ライブラリは、別のアカウントからでも利用することができますので、自身が公開することで誰かに利用してもらうことができますし、誰かが公開していれば、それを利用することもできます。

### 【ライブラリの使用方法】

1. 使用したい関数の含まれているプロジェクトの設定画面から、スクリプトIDをコピーします。



Apps Script 無題のプロジェクト デプロイ ▼

① プロジェクトの設定

<>

🕒 全般設定

⋮

⚙️

Apps Script プロジェクト全体に関する設定です。これらの設定を変更しても既存のデプロイメントは影響を受けません。

- ☒ キャッチされなかった例外を Cloud のログに記録する
- ☒ Chrome V8 ランタイムを有効にする
- ☐ 「appscript.json」 マニフェスト ファイルをエディタで表示する

ID

ID は、Apps Script プロジェクトの一意の識別子です。

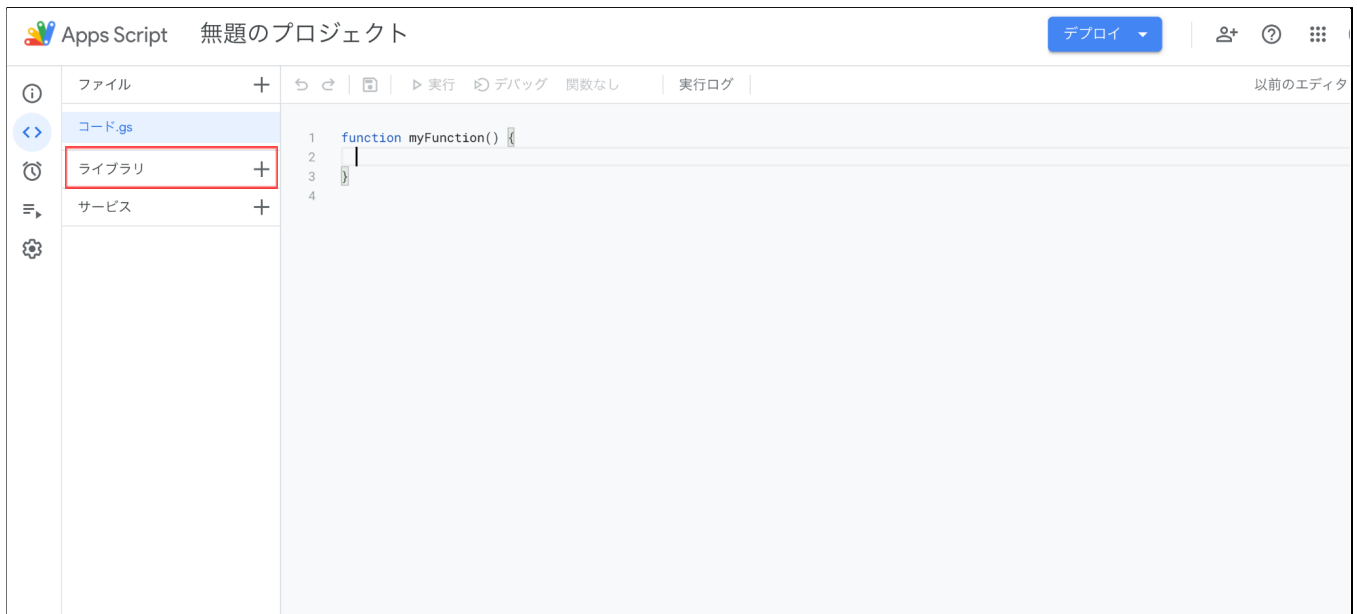
スクリプト ID	1IGpxC4N6F
----------	------------

[コピー](#)

Google Cloud Platform (GCP) プロジェクト

Apps Script プロジェクトでは、Google Cloud Platform を使用して承認、高度なサービス、その他の詳細が管理されます。詳しくは、[Google Cloud Platform](#) をご覧ください。

2. ライブラリを追加したいプロジェクトを開き、左のメニューバーの「ライブラリ +」を選択します。



3. コピーしてきたスクリプトIDを貼り付け、任意のIDを設定します。ここでは、testとします。



4. 追加したライブラリは、「ライブラリ +」の下にID名で表示されるので、誤りがないか確認します。



5. 「ID.関数()」と記載することで、追加したライブラリ内の関数を使用することができます。以下の例では、「test.myFunction()」と記載し、「Hello, GAS」とログを表示したmyFunction関数を使用しています。



## ■ 公式リファレンスとは

### 【公式リファレンスの概要】

公式リファレンスはプログラミング言語ごとに存在する、その公式組織が出している言語の説明書です。  
GASのリファレンスは英語で書かれており、どこよりも詳しくかつ正確な情報が細かく載っているのが特徴です。

GASの公式リファレンス <https://developers.google.com/apps-script/reference>

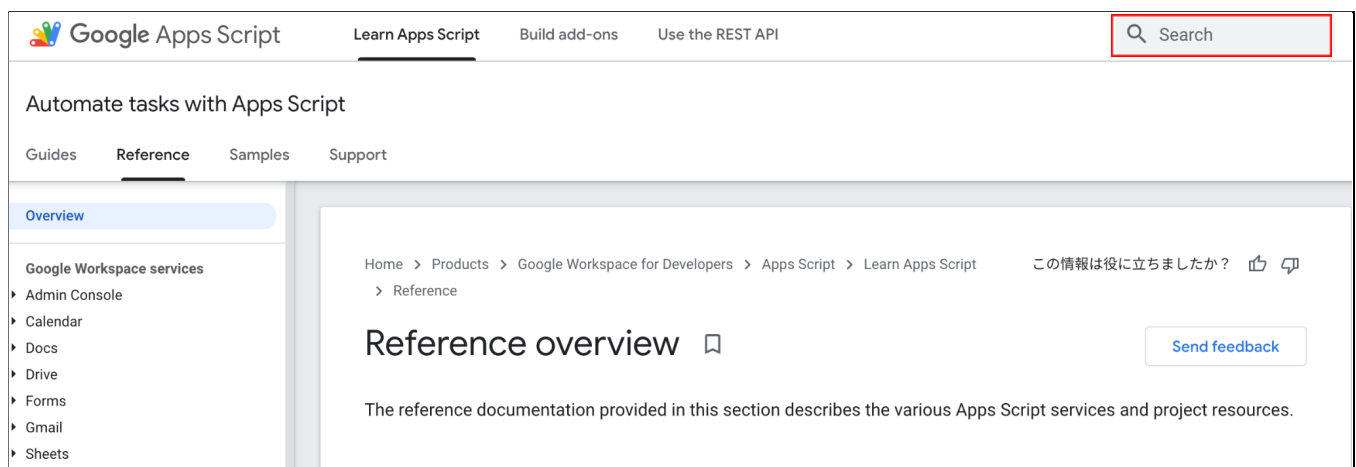
参考: 公式リファレンスの検索ワード「gas app script reference」

### 【公式リファレンスの活用方法】

公式リファレンスの活用方法は様々ありますが、本教材では2つの方法を紹介します。

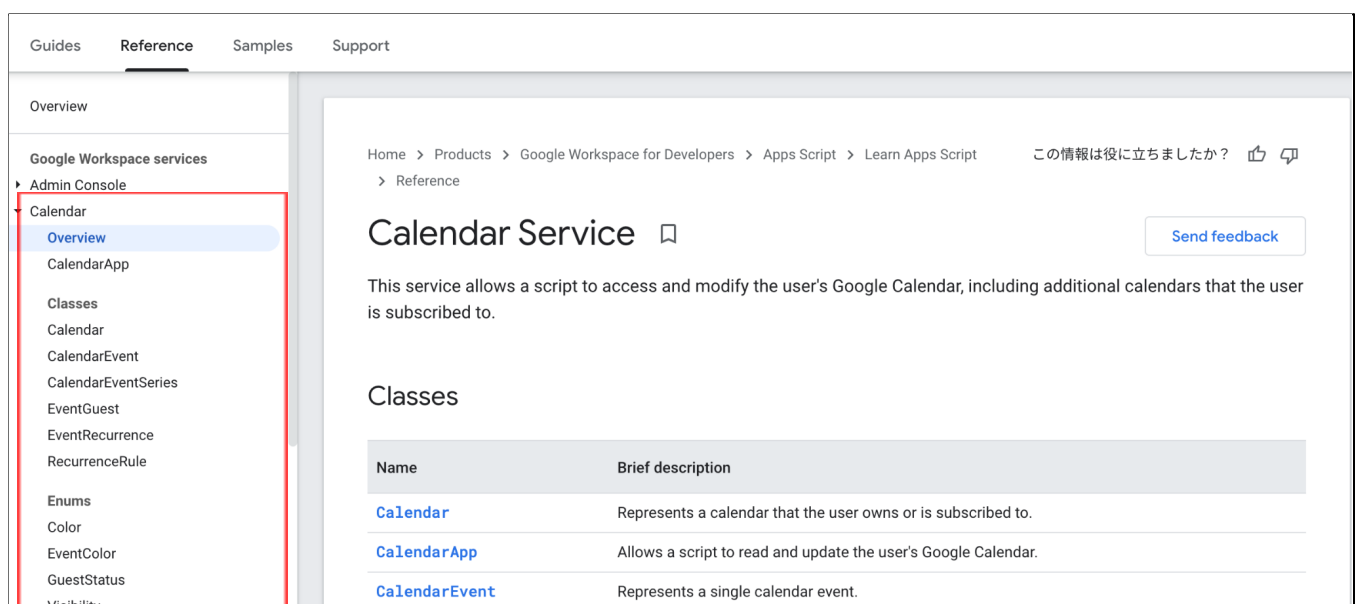
#### < 気になった用語を検索 >

気になった用語が出てきた場合は、画面右上の「Search」に入力して、検索することができます。



#### < 特定サービスの内容を深掘りする方法 >

ページ左側のメニューバーから、特定のサービスを選択し、そのサービスで利用できるクラス (Classes) やプロパティ (Properties)、メソッド (Methods)を確認することができます。例えば、Googleカレンダーのサービスについて知りたいと思った時は、「Calendar」の「Overview」をクリックします。



ページを下にスクロールすると、それぞれのクラスで利用できるプロパティ(Properties)やメソッド(Methods)について記載があります。

Overview

Google Workspace services

Admin Console

Calendar

Overview

CalendarApp

Classes

Calendar

CalendarEvent

CalendarEventSeries

EventGuest

EventRecurrence

RecurrenceRule

Enums

Color

EventColor

GuestStatus

Visibility

Advanced services

Calendar API

Docs

Drive

Forms

Gmail

Sheets

CalendarApp

Properties

Property	Type	Description
Color	Color	An enum representing the named colors available in the Calendar service.
EventColor	EventColor	An enum representing the named event colors available in the Calendar service.
GuestStatus	GuestStatus	An enum representing the statuses a guest can have for an event.
Month	Month	An enum representing the months of the year.
Visibility	Visibility	An enum representing the visibility of an event.
Weekday	Weekday	An enum representing the days of the week.

Methods

Method	Return type	Brief description
<a href="#">createAllDayEvent(title, date)</a>	CalendarEvent	Creates a new all-day event.
<a href="#">createAllDayEvent(title, startDate, endDate)</a>	CalendarEvent	Creates a new all-day event.
<a href="#">createAllDayEvent(title, startDate, endDate, options)</a>	CalendarEvent	Creates a new all-day event.

さらに詳しく調べたい場合は、青字部分をクリックすると詳細(Detailed documentation)へ遷移することができます。以下の例は、「[createAllDayEvent\(title, date\)](#)」をクリックしたケースです。

Overview

Google Workspace services

Admin Console

Calendar

Overview

CalendarApp

Classes

Calendar

CalendarEvent

CalendarEventSeries

EventGuest

EventRecurrence

RecurrenceRule

Enums

Color

EventColor

GuestStatus

Visibility

Advanced services

Calendar API

Docs

Drive

Forms

Gmail

Sheets

Slides

More...

Detailed documentation

createAllDayEvent(title, date)

Creates a new all-day event.

```
// Creates an all-day event for the moon landing and logs the ID.
var event = CalendarApp.getDefaultCalendar().createAllDayEvent('Apollo 11 Landing',
    new Date('July 20, 1969'));
Logger.log('Event ID: ' + event.getId());
```

Parameters

Name	Type	Description
title	String	the title of the event
date	Date	the date of the event (only the day is used; the time is ignored)

Return ↗

CalendarEvent — the created event

Authorization

Scripts that use this method require authorization with one or more of the following [scopes](#) or appropriate scopes from the [related REST API](#):