


指導資料

情報教育 第157号

 鹿児島県総合教育センター
令和3年10月発行

対象校種
幼稚園 小学校 中学校
義務教育学校 高等学校
特別支援学校



クラウドサービスをより効率よく利用するために ～「GAS」による作業の自動化・効率化～

「GIGAスクール構想」の実現に向けてICT環境が整いつつある現在、各種クラウドサービス*¹利用への関心が高まるとともに、授業や校務での様々な活用が期待される。そこで、本稿では、業務効率を高める一つの手段として、プログラミング言語「GAS」の活用を紹介する。

1 はじめに

令和2年度に、本県の公立学校の教員及び児童生徒に一人1アカウントが付与され、MicrosoftとGoogleの両方のクラウドサービスの利用が可能となった。インターネットが使える環境であれば、いつでもどの端末からでも自分のデータにアクセスすることができ、児童生徒の学習や教員の業務改善に向けた様々な活用が期待される。

このクラウドサービスでは、児童生徒の個別最適な学びを実現するデジタルコンテンツや協働的な学びの実現に効果的な共同編集が可能なアプリケーション、そして教員の業務を効率化するアプリケーションなどが数多く提供されており、すでに、欠席・遅刻の連絡や各種調査の回答をアンケートフォームで送信・自動集計することで大幅な業務改善につながったという事例が見られる。一方で、授業や校務においてクラウドサービス活用のための準備や運用が新たな業務として発生している。また、クラウドサービスの中には同じ作業を繰り返す必要があるなど効率の悪いものもあるため、負担に感じる教員も多いのではないだろうか。

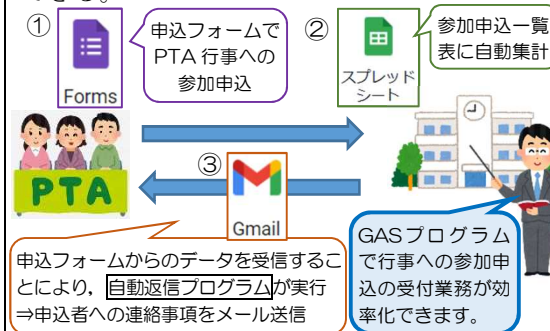
そこで、今回は、Googleが提供する複数のクラウドサービスを連携することができるプログラミング言語「GAS」の活用方法を示し、業務効率を高める一つの手段として紹介する。

2 「GAS」とは

「GAS」とは「Google Apps Script」の頭文字を取った略称である。また、「GAS」のベースはJavaScript*²という様々な用途に活用されているプログラミング言語である。GASの特長は次のように示すことができる。

資料1 「GAS」の特長

- ・ スクリプト言語という簡易に記述や実行ができるプログラミング言語であるため、習得が容易で活用の場が多い。
- ・ Googleアカウントがあれば無償で利用でき、開発環境を構築する必要がないため、気軽に取り組むことができる。
- ・ Gmailやフォームなど、Googleが提供するサービスや、Google以外の外部アプリケーションについても連携して活用することができる。



*1) サーバやソフトウェアなどコンピュータに必要な機能を、ネットワークを経由して、サービスの形で提供する利用形態
 *2) 主に Web ページに組み込まれたプログラムを Web ブラウザ上で実行するために用いられるプログラミング言語の一つ

「GAS」を利用するには、Googleが提供するスクリプトエディタで目的に応じたプログラムを作成する必要がある。スクリプトエディタを起動する方法は次の2通りである。

資料2 「GAS」利用のための準備

方法1

- ① Google Chromeを起動、Googleアプリランチャー (☰) をクリックし、開いたウィンドウの中から「スプレッドシート」を選択し、新しいスプレッドシートを作成する。
- ② 「ツール」をクリックし、「スクリプトエディタ」を選択して起動する。

方法2

- ① Google Chromeを起動、Googleアプリランチャー (☰) をクリックし、開いたウィンドウの中から「ドライブ」をクリックする。
- ② 「マイドライブ」から、「その他」を開く。
- ③ 「Google Apps Script」をクリックして起動する。

※ 「その他」に「Google Apps Script」が表示されない場合は、「アプリを追加」から「Google Apps Script」を選択する。

3 「GAS」プログラミングの基礎

(1) 「GAS」プログラミングのルール

ア プログラムの記述は英数文字を使用する(大文字と小文字は区別される。)

イ 処理の最小単位をステートメントと呼び、末尾に「;」を付ける。

ウ 一連の処理をまとめたものを関数と呼び、まとめる範囲(ブロック)を {} で囲む。ブロックは条件分岐や反復などの制御フロー構文*3でも用いる。

エ {} の対応とブロックの多重構造を見やすくするための改行や字下げ(インデント)、空白(スペース)の挿入を行うかは基本的に自由である。

オ //以降はコメント(注釈文)とみなされ処理には影響しない。複数行をまたぐ場合は「/*」と「*/」で囲む。

カ オブジェクト指向*4であり、外部ライブラリ*5や組み込み関数*6、自作した関数が使用できる。

(2) プログラムの記述例

```
//合格判定
function seiseki() {
  var ten = Browser.inputBox("あなたの点数は");
  if(ten > 80){
    Browser.msgBox("合格");
  }
  else{
    Browser.msgBox("不合格");
  }
}
```

このコード例には、以下の注釈が付けられています:

- コメント: //合格判定
- 関数名: function seiseki()
- ブロック文の始まり: {
- 入力ボックスの表示: Browser.inputBox("あなたの点数は");
- メッセージボックスの表示: Browser.msgBox("合格");
- ブロック文の終わり: }

※ inputBoxメソッド*7・msgBoxメソッドを用いるプログラムにおいては、スクリプトエディタの起動は方法1で行う。

(3) プログラムの実行手順

ア スクリプトエディタを起動する。

イ プロジェクト*8の名前を変更する。

ウ プログラムを入力、保存する。

※ 他のGoogleサービスと異なり、自動保存されないため注意する。

エ 「実行」をクリックする。実行ログが表示される。(2)のプログラムであれば、inputBoxとmsgBoxはスプレッドシート上に表示される。

※ プログラムの実行の際に、アカウントの確認や安全なファイルであるかどうかの確認を求められる。

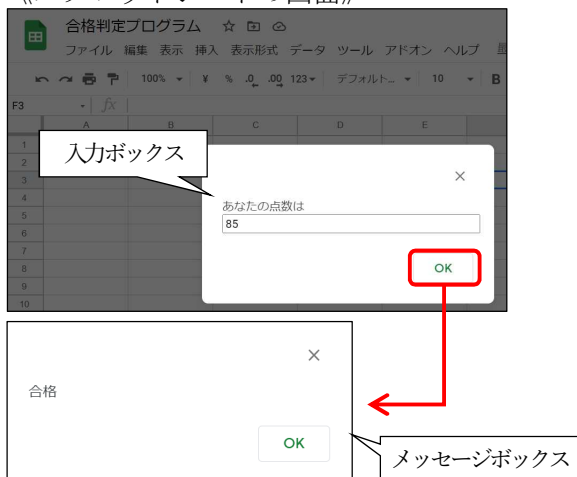
実行結果

《スクリプトエディタの画面》



*3) プログラムの実行の流れを制御する文 *4) データと手続きをオブジェクトと呼ばれる一つのまとまりとして定義し、オブジェクトを組み合わせるシステムを構築していく手法 *5) 外部から呼び出して利用できる部品化されたプログラムを集めたもの *6) プログラミング言語などの仕様にあらかじめ用意され、標準で使用できる関数

《スプレッドシートの画面》



(4) プログラミングの命令コード

「GAS」の演算子や制御文などの命令コードは、他のプログラミング言語と似ているものが多いため、プログラミングの経験があれば命令コードに違和感を感じることは少ない。命令コードの詳細については省略するので、文部科学省の高等学校「情報 I」教員研修用教材《JavaScript版》や関係書籍・Web サイトを参考にしてほしい。また、VBA (Visual Basic Applications) *9で用いられるアルゴリズムも参考にするとよい。

(5) 「GAS」の基本的なプログラム

MicrosoftのExcelやPowerPointなどの操作を自動化するVBAと同様に、「GAS」はGoogleの様々なクラウドサービスの操作を自動化することができる。

ア 新しいスプレッドシート (ファイル名「newSheet」) を作成するプログラム

```
function myFunction() {
  SpreadsheetApp.create("newSheet");
}
```

※ ファイルはGoogleドライブに格納される。

イ 編集中的スプレッドシートの指定したセルに、文字を表示するプログラム

```
function myFunction() {
  // アクティブ状態のスプレッドシートを取得
  var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
  // アクティブ状態のシートを取得
  var sheet = spreadsheet.getActiveSheet();
  // セルの指定
  var cell = sheet.getRange("C2");
  // 値の設定
  cell.setValue("Hello world!");
}
```

※ getRange("C2")は、getRange(2, 3)と記述しても同様のセルの指定ができる。

実行結果

	A	B	C
1			
2			Hello world!

ウ スプレッドシートの指定したセルの値を取得するプログラム

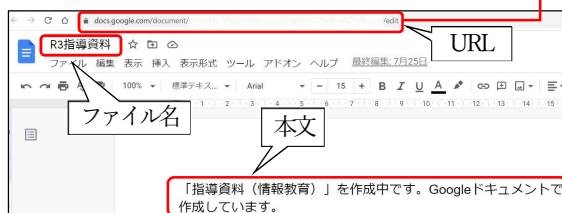
```
function myFunction() {
  // アクティブ状態のスプレッドシートを取得
  var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
  // アクティブ状態のシートを取得
  var sheet = spreadsheet.getActiveSheet();
  // セルの範囲指定
  var range = sheet.getRange("B1:B5");
  // 値の取得
  var values = range.getValues();
  // ログに出力
  console.log(values);
}
```

実行結果

	A	B
1	1	相川○○
2	2	井原△△
3	3	内野○○
4	4	加藤◇◇
5	5	佐藤△△

エ ドキュメントの文書を取得するプログラム

```
function logDocument() {
  // 取得するドキュメントの URL を指定
  const DOC_URL =
    "https://docs.google.com/document/d/1w1MydWwMj7lwmragkFhoPtyV0VSwi8cwSC9cPlw-x/edit";
  const doc = DocumentApp.openByUrl(DOC_URL);
  // ドキュメントのファイル名をログに出力
  console.log(doc.getName());
  // ドキュメントの本文をログに出力
  console.log(doc.getBody().getText());
}
```



実行結果

	ファイル名	本文
22:13:58	R3指導資料	指導資料 (情報教育) を作成中です。Googleドキュメントで作成しています。

*7) オブジェクトにおける手続 (=処理) の要素 メソッドを呼び出す構文:「オブジェクト名.メソッド名(引数)」

*8) スクリプト (スクリプト言語で記述したプログラム) の作成単位であり、単一または複数のファイルで構成される

*9) マクロを作成するための Microsoft 社のプログラミング言語。マクロとは一連の手順を踏む作業を登録することで自動実行する機能

4 「GAS」を活用したクラウドサービスの連携の実例

「GAS」を活用すれば、Googleが提供する複数のクラウドサービスを連携させることができる。

(1) フォームとGmailの連携

例えばGoogleフォームで送信されてきた申込を受け付けた際、Gmailと連携し、メールアドレスへ自動返信するプログラムを作成することができる。方法は以下のとおりである。

参加申込フォーム

PTA研修観察に参加される方は、氏名とメールアドレスを入力して送信をお願いします。
*必須

氏名*

回答を入力

	A	B	C
1	タイムスタンプ	氏名	メールアドレス
2	2021/07/26 0:51:01	青山 ○○	*****@kago.ed.jp
3	2021/07/26 0:59:07	加藤 ○○	*****@kago.ed.jp
4	2021/07/26 1:01:21	田中 ○○	*****@kago.ed.jp

メールアドレス*

回答を入力

※ フォームからの回答を集計するスプレッドシートを作成後、スプレッドシートにおいてスクリプトエディタを起動する。

```
function Form(e) {
  // フォームの回答を取得
  var name = e.namedValues["氏名"][0];
  var email = e.namedValues["メールアドレス"][0];
  // 自動返信メール件名
  var subject = "参加申込について";
  // 自動返信メール本文
  var body = name + "様" + "\n" +
    "参加申し込み、ありがとうございます。" +
    "当日は午前9時に体育館前へご集合下さい。"
  // メール送信
  MailApp.sendEmail(email, subject, body);
}
```

実行結果

参加申込について

***** @kago.ed.jp

To 自分

青木誠様

参加申し込み、ありがとうございます。
当日は午前9時に体育館前へご集合下さい。

※ フォームの送信をトリガー*10にして、プログラムが実行されるように設定する。

Apps Script Gmail自動返信

概要

エディタ

トリガー

実行数

プロジェクトの設定

Gmail自動返信のトリガーを編集

実行する関数を選択

Form

エラー通知設定

毎日通知を受け取る

デプロイ時に実行

Head

イベントのソースを選択

スプレッドシートから

イベントの種類を選択

フォーム送信時

(2) フォームとスプレッドシートの連携
スプレッドシートの問題データとGoogleフォームを連携し、テスト形式のフォームを自動作成するプログラムを作ることができる。方法は以下のとおりである。

テスト作成用 (提供データ)

番号	問題	選択肢1	選択肢2	選択肢3	解答	配点
1	コンピュータ機器やシステムなどの、購入から運用、廃棄までにかかる費用の総額。	総保有コスト	イニシャルコスト	ランニングコスト	1	25
2	指定システム設備の記録情報であり、木の非輪のように同心円状に形成された組織。	セクタ	シリンドラ	トラック	3	25
3	階層構造でファイルを管理する場合、階層の最上位にあるディレクトリ。	サブディレクトリ	ルートディレクトリ	カレントディレクトリ	2	25

```
function testform() {
  //スプレッドシートの問題データを取得
  var ActiveSheet = SpreadsheetApp.getActiveSheet();
  var lastRow = ActiveSheet.getLastRow();
  var lastColumn = ActiveSheet.getLastColumn();
  var Testlist = [];
  Testlist = ActiveSheet.getRange(3, 2, lastRow-2, lastColumn-1).getValues();
  //セルC1に入力したタイトルでテスト形式のフォームを生成
  var formTitle = ActiveSheet.getRange(1, 3).getValue();
  var form = FormApp.create(formTitle);
  form.setIsQuiz(true);
  //最終問題まで問題 (選択式:チェックボックス) を追加
  for(var i = 0; i<Testlist.length; i++) {
    var choices;
    var item = form.addMultipleChoiceItem();
    //選択肢の作成、正解の設定
    choices=[
      item.createChoice(Testlist[i][1], Testlist[i][4]=1),
      item.createChoice(Testlist[i][2], Testlist[i][4]=2),
      item.createChoice(Testlist[i][3], Testlist[i][4]=3)
    ];
    //各設問の見出し、選択肢、配点、必須を設定
    item.setTitle([Testlist[i][0]]);
    item.setChoices(choices);
    item.setPoints(Testlist[i][5]);
    item.setRequired(false);
  }
}
```

実行結果

「情報処理」用語問題

フォームの説明

コンピュータ機器やシステムなどの、購入から運用、廃棄までにかかる費用の総額。

総保有コスト

イニシャルコスト

ランニングコスト

5 おわりに

本稿では自動返信とテスト作成という業務の効率化に有効と考えられる活用例を示した。これにドキュメントやスプレッドシートからのデータ取得を組み合わせるだけでも活用の幅は広がる。また、「GAS」のベースになっているJavaScriptは高等学校「情報 I」の学習言語として注目されている。環境構築が不要ですぐに始められるので、業務や授業に活用されることを期待している。

一引用・参考文献一

- 文部科学省「高等学校情報科『情報 I』教員研修用教材」平成 31 年 3 月 29 日

(情報教育研修課 青木 誠)

* 10) 特定のイベントに合わせて自動的にプログラムを実行させる機能