報道各位

impress R&D
An impress Group Company

2017年9月8日
株式会社インプレスR&D
http://nextpublishing.jp/

「技術書典シリーズ」第三弾！
『サーバーレスでお手軽自動化！Google Apps Script 活用入門』発行
コピペでそのまま使える！　サンプルコード満載

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『サーバーレスでお手軽自動化！Google Apps Script 活用入門』（著者：中山 貴幸）を発行いたします。

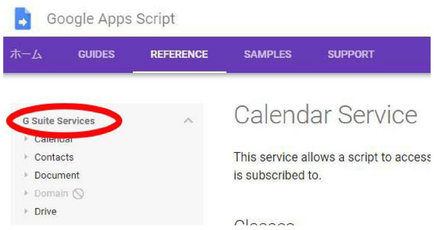
『サーバーレスでお手軽自動化！Google Apps Script活用入門』
http://nextpublishing.jp/isbn/9784844397908

著者：中山 貴幸
小売希望価格：電子書籍版　1000円（税別）／印刷書籍版　1500円（税別）
電子書籍版フォーマット:EPUB3／Kindle Format8
印刷書籍版仕様：B5判／モノクロ／本文102ページ
ISBN：978-4-8443-9790-8
発行：インプレスR&D

＜＜発行主旨・内容紹介＞＞
【コピペでお手軽自動化！GoogleAppsScriptの初歩から活用までをサクッと紹介！】

Gmail や Google ドライブ、Google スプレッドシートなどを使って身の回りの作業を簡単に自動化できるGoogleAppsScript（GAS）の活用入門書です。
GASの使い方から簡単なスクリプト、少し高度な使い方まで豊富なソースコードを多数収録。様々な作業を自動化・効率化するためのノウハウをわかりやすく紹介しています。
（本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。）

GASの機能をわかりやすく図表とコードで丁寧に解説



外部ドメインにアクセスしてみよう

GASの魅力的な機能には、JavaScriptでは扱えないものがたくさんあります。その中でもUrlFetchAppというものについて深く学んでいきましょう。

ScriptServiceを用いてGAS特有の処理を扱おう

JavaScriptでは扱えない、ScriptServiceもGASの魅力の一つです。これは、Googleのサーバー上で動くからこその機能です。

この章では、主に**`UrlFetchApp`**という機能に絞って説明をしていきます。他の機能については、ぜひ自分で調べてみてください。

外部のドメインにあるリソースを取得してみよう

GASでは`UrlFetchApp`を利用して、簡単に外部ドメインへのアクセスができます。例をみながら実際に扱えるようにしていきましょう。

コード2.3

```
function myFunction(){
    //URLからリソースを受け取ります
    var html = getResource("http://takanakahiko.me/test.html");

    //取得した内容をログに出力します。
    Logger.log(html);
  }
```



```
function getResource(url) {
 //urlにアクセスした結果をresponseに格納します。
  var response = UrlFetchApp.fetch(url);
  //response内にあるcontentTextを文字列型で取り出します。
  var contentText = response.getContentText();
  //結果を返します
  return contentText;
}
```

では、上記ソースコードを実行しましょう。

ログを開いてみてください。以下のようになっていれば成功です。

myFunction()を実行したあとのログ

```
<!DOCTYPE html>
 <html>
   <head>
     <meta charset="utf-8">
     <title>テスト</title>
   </head>
   <body>
     これはテストです
   </body>
 </html>
```

つまり、外部のWebサイト等のリソースを取得できるということです。

これを使うことで、様々な利点を得ることができます。その一例として、WebAPIを使用する例を挙げていきます。

WebAPIを利用してみよう

WebAPIは、簡単に言うと**外部のサービスを利用することができる仕組み**です。WebAPIは主に、JSONかXMLという形式が用いられます。今回はJSONの形式で配信されるものを利用してみましょう。

ここでは**WebAPIから天気を取得する**といった実例で解説します。

WebAPIを選ぼう

今回は天気の取得をしていきたいので、天気関連のWebAPIが必要です。試しにGoogleで「天気予報 web api json」と調べてみたら、livedoorの「お天気Webサービス」がヒットしたので、それを使ってみましょう。

G SuiteとGASの連係は特に細かく紹介

Googleスプレッドシートの操作

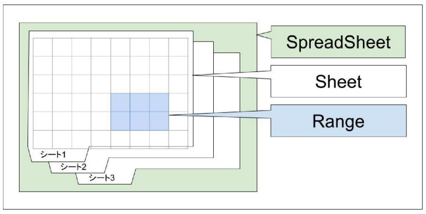
GASでスプレッドシートを操作する際に、3つのデータ構造（型のようなもの）を意識することになります。

SpreadSheet：一つのスプレッドシートファイルのことです。複数のシートを持ちます。

Sheet：SpreadSheet内にあるシートのことです。

Range：Sheet内にあるマス目の集合のことです。

Spreadsheet と Sheet、Range のイメージ



つまり、データを出し入れするということは、Rangeを取得し操作する、といえます。

シートの取得

SpreadSheetの取得、更にSheetの取得を行うソースコードです。

`sheetId`は、実際にスプレッドシートを開いて、そのURLから確認できます。下の例では`XXXXX`が`id`になります。

```
https://docs.google.com/spreadsheets/d/XXXXX/edit
```

以下のコードでは、`id`が`XXXXX`であるSpreadSheet内の、シート1といったSheetを取得しています。

コード3.4

```
var sheetId = "XXXXX";
```



```
var sheet = SpreadsheetApp.openById(sheetId).getSheetByName("シート
1");
```

2次元配列でのデータ取得

シート内のRangeの取得では、シート内の全ての値や一部の領域を2次元配列として取得することができます。

一部の領域を取得する際の変数

変数	内容
row	上からのマス数（y座標）
column	左からのマス数（x座標）
numRow	取得したい範囲の縦幅（height）
numColumns	取得したい範囲の横幅（width）

次のコードには2つの例がありますが、前者ではシート内のすべての値を取得しています。後者の場合は、この表の変数を使って範囲指定をしています。

コード3.5

```
var sheetDataArray1  = sheet.getDataRange().getValues(); //全ての値
var sheetDataArray2  = sheet.getRange(row, column, numRows,
numColumns).getValues(); //指定した範囲の値
```

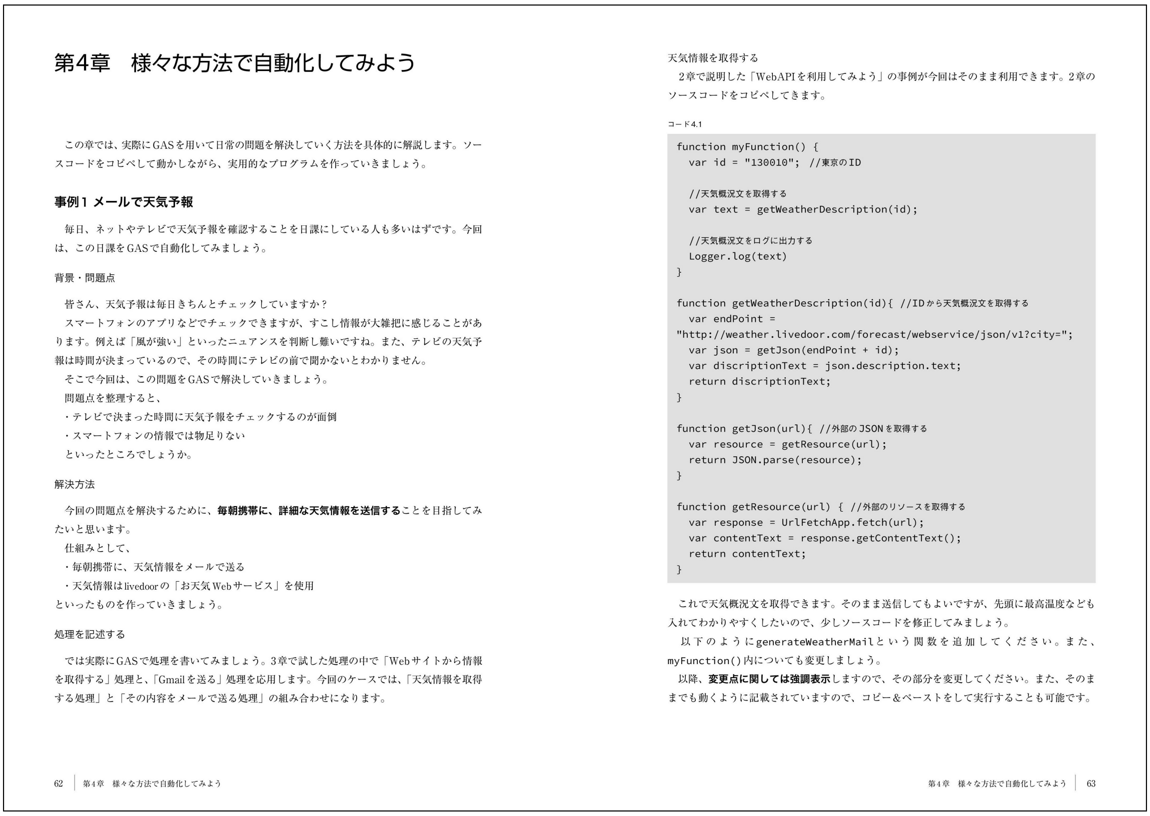
2次元配列でのデータセッティング

シート内のRangeを取得し、そこへデータをセットします。セットするデータは2次元配列（例では`sheetDataArray`）を用います。引数は以下のように設定します。

変数	内容
row	上からのマス数（y座標）
column	左からのマス数（x座標）
numRow	取得したい範囲の縦幅（height）
numColumns	取得したい範囲の横幅（width）

```
sheet.getRange(row, column,
numRows,numColumns).setValues(sheetDataArray);
```

紹介した機能を使ったGASの利用事例を、コードを含めて紹介



第4章　様々な方法で自動化してみよう

この章では、実際にGASを用いて日常の問題を解決していく方法を具体的に解説します。ソースコードをコピペして動かしながら、実用的なプログラムを作っていきましょう。

事例1 メールで天気予報

毎日、ネットやテレビで天気予報を確認することを日課にしている人も多いはずです。今回は、この日課をGASで自動化してみましょう。

背景・問題点

皆さん、天気予報は毎日きちんとチェックしていますか？
スマートフォンのアプリなどでチェックできますが、すこし情報が大雑把に感じることがあります。例えば「風が強い」といったニュアンスを判断し難いですね。また、テレビの天気予報は時間が決まっているので、その時間にテレビの前で聞かないとわかりません。
そこで今回は、この問題をGASで解決していきましょう。
問題点を整理すると、
・テレビで決まった時間に天気予報をチェックするのが面倒
・スマートフォンの情報では物足りない
といったところでしょうか。

解決方法

今回の問題点を解決するために、**毎朝携帯に、詳細な天気情報を送信する**ことを目指してみたいと思います。
仕組みとして、
・毎朝携帯に、天気情報をメールで送る
・天気情報はlivedoorの「お天気Webサービス」を使用
といったものを作っていきましょう。

処理を記述する

では実際にGASで処理を書いてみましょう。3章で試した処理の中で「Webサイトから情報を取得する」処理と、「Gmailを送る」処理を応用します。今回のケースでは、「天気情報を取得する処理」と「その内容をメールで送る処理」の組み合わせになります。

62　第4章　様々な方法で自動化してみよう

天気情報を取得する
2章で説明した「WebAPIを利用してみよう」の事例が今回はそのまま利用できます。2章のソースコードをコピペしてきます。

コード4.1

```
function myFunction() {
  var id = "130010";  //東京のID

  //天気概況文を取得する
  var text = getWeatherDescription(id);

  //天気概況文をログに出力する
  Logger.log(text)
}

function getWeatherDescription(id){ //IDから天気概況文を取得する
  var endPoint =
"http://weather.livedoor.com/forecast/webservice/json/v1?city=";
  var json = getJson(endPoint + id);
  var discriptionText = json.description.text;
  return discriptionText;
}

function getJson(url){ //外部のJSONを取得する
  var resource = getResource(url);
  return JSON.parse(resource);
}

function getResource(url) { //外部のリソースを取得する
  var response = UrlFetchApp.fetch(url);
  var contentText = response.getContentText();
  return contentText;
}
```

これで天気概況文を取得できます。そのまま送信してもよいですが、先頭に最高温度なども入れてわかりやすくしたいので、少しソースコードを修正してみましょう。
以下のようにgenerateWeatherMailという関数を追加してください。また、myFunction()内についても変更しましょう。
以降、**変更点に関しては強調表示**しますので、その部分を変更してください。また、そのままでも動くように記載されていますので、コピー&ペーストをして実行することも可能です。

第4章　様々な方法で自動化してみよう　63

＜＜目次＞＞

第1章　GoogleAppsScriptを始めよう
　GoogleAppsScriptでできること
　開発の準備をしよう
　Hello World! on GoogleAppsScript
第2章　GAS特有の機能を覚えよう
　GASで扱えるGoogleの11サービス
　外部ドメインにアクセスしてみよう
　GASの豊富な実行トリガー
第3章　よく使う処理を試してみよう
　変数の状態を確認する
　Googleスプレッドシートの操作
　Gmailの操作
　Googleドキュメントの操作
　Googleフォームからの起動・値の取得
　GETやPOSTの受け取り
　外部JSON(XML)の取得
　ライブラリのインポート
　Twitterとの通信
　Slackとの通信
第4章　様々な方法で自動化してみよう
　事例1　メールで天気予報
　事例2　議事録の自動生成

事例３　画像の一括処理
事例４　Twitter タイムラインの監視

＜＜著者紹介＞＞
中山 貴幸
1996 年生まれ、東京都出身。拓殖大学工学部在学。在学中は組込系開発会社にて各種プロジェクトに従事。C++ / Python / JavaScript / AndroidJava などを得意とする。

＜＜販売ストア＞＞
電子書籍:
Amazon Kindle ストア、楽天 kobo イーブックストア、Apple iBookstore、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER
印刷書籍:
Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス
※　各ストアでの販売は準備が整いしだい開始されます。
※　全国の一般書店からもご注文いただけます。

【株式会社インプレス R&D】 http://nextpublishing.jp/
株式会社インプレス R&D（本社：東京都千代田区、代表取締役社長：井芹昌信）は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。
※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム（またはメソッド）の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド（POD）による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 http://www.impressholdings.com/
株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:唐島夏生、証券コード:東証１部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「モバイルサービス」を主要テーマに専門性の高いコンテンツ＋サービスを提供するメディア事業を展開しています。2017 年 4 月 1 日に創設 25 周年を迎えました。

【お問い合わせ先】
株式会社インプレス R&D NextPublishing センター
〒101-0051　東京都千代田区神田神保町 1-105
TEL 03-6837-4820
電子メール: np-info@impress.co.jp