```
ここでは、以下のSheetクラスのメソッドについて説明します。
    メソッド
                                            説明
                                   戻り値の型
    getRange(a1Notation)
                                   Range
                                            A1表記でセルの範囲を取得
                                            行番号・列番号指定でセルの範囲を取得
    getRange(row, column, numRows, numColumns)
                                   Range
                                            データが存在するセルの範囲を取得
    getDataRange()
                                   Range
   getRange は2種類記載しましたが、セルの範囲の書き方に違いがあるだけです。使い方は後ほど説明します。まずは、使い
   やすいほうを使ってください。
   Sheetクラスのメソッド一覧については公式サイトに記載されているため、興味のある方は参照してみてください。
   getRange(a1Notation)
   getRange(a1Notation)は、A1表記(A1 notation)でセルの範囲を取得するメソッドです。
   A1表記とは、「A1:C5」のように範囲を指定する表記方法です。この場合A1からC5までの範囲を表します。
   getRange(a1Notation) は、以下のように使用します。
   スクリプトファイル(見本)
 1 let range = SpreadsheetApp.getActiveSheet().getRange('A1:C5');
   それでは、実際にスクリプトを記述してみましょう。ここではgetRange()で取得した範囲の値を出力します。
   その前に準備として、以下のように「シート1」に書きましょう。
                                                  E
    2
                         5
                         8
                                  12
               10
                        11
                                  15
     5
               13
                        14
    7
     8
     9
    10
   シートの準備ができたら、スクリプトを記述します。関数名は自由ですが、ここでは「mySheet」を新たに作成し、以下
   のように実装します。
    1. アクティブなシートの取得
    2. A1表記でセルの範囲を取得
    3. セルの範囲の値をコンソールログに出力
   GAS入門
  1 + function mySheet() {
  3 + let sheet = SpreadsheetApp.getActiveSheet();
  4+
  5 + // A1表記でセルの範囲を取得
  6 + let range = sheet.getRange('A1:C3');
  7 + console.log(range.getValues());
  8+}
   スクリプトを保存して「mySheet」を実行すると、以下のように表示されます。
 1 [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
   取得した範囲(A1:C3)の値が表示されていることがわかります。
   getRange(row, column, numRows, numColumns)
   getRange(row, column, numRows, numColumns)は行番号・列番号指定でセルの範囲を取得するメソッドです。
   getRange(row, column, numRows, numColumns) は以下のように使用します。
   スクリプトファイル(見本)
 1 let range = SpreadsheetApp.getActiveSheet().getRange(1, 1, 4, 3);
   この場合、1行目1列目(A1)から4行3列分(C4)の範囲を取得します。つまり、A1:C4の範囲が取得できるということで
   す。それでは、実際にスクリプトを記述してみましょう。
   ここでは、getRange(row, column, numRows, numColumns)で1行目1列目(A1)から4行3列分(C4)の範囲を取得し、値を
   コンソールログに表示します。
   GAS入門
  1 function mySheet() {
  2 // アクティブな(現在開いている)シートの取得
     let sheet = SpreadsheetApp.getActiveSheet();
  5 // A1表記でセルの範囲を取得
  6 let range = sheet.getRange('A1:C3');
  7 console.log(range.getValues());
  8 +
  9+ // 行番号・列番号指定でセルの範囲を取得
 10 + range = sheet.getRange(1, 1, 4, 3);
 11 + console.log(range.getValues());
 12 }
   スクリプトを保存して「mySheet」を実行すると、以下のように表示されます。
 1 [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
 2 [ [ 1, 2, 3 ], [ 4, 5, 6 ], [ 7, 8, 9 ], [ 10, 11, 12 ] ]
   詳細は以下のとおりです。
    ■ 1行目:AI表記の getRange() で取得した範囲の値(A1:C3)を表示
    ■ 2行目: 行番号・列番号指定の getRange() で取得した範囲の値(A1:C4)を表示
   getDataRange()
   getDataRange()は、データが存在するセルの範囲を取得するメソッドです。 getDataRange() は、以下のように使用しま
   す。
   スクリプトファイル(見本)
 1 let range = SpreadsheetApp.getActiveSheet().getDataRange();
   シートが以下の状態の場合、A1:C5の範囲を取得します。
                                                  E
                     В
                              C
                                        D
                         2
               1
    3
                         8
                                  12
                        11
               10
               13
                                  15
                        14
    7
     8
     9
    10
   それでは、実際にスクリプトを記述してみましょう。 getDataRange() でデータ範囲を取得して、値をコンソールログに表
   示します。
   GAS入門
   1 function mySheet() {
   2 // アクティブな(現在開いている)シートの取得
      let sheet = SpreadsheetApp.getActiveSheet();
      // A1表記でセルの範囲を取得
       let range = sheet.getRange('A1:C3');
       console.log(range.getValues());
       // 行番号・列番号指定でセルの範囲を取得
       range = sheet.getRange(1, 1, 4, 3);
       console.log(range.getValues());
      // データが存在するセルの範囲を取得
       range = sheet.getDataRange();
      console.log(range.getValues());
   16 }
   スクリプトを保存して「mySheet」を実行すると、以下のように表示されます。
 1 [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
 2 [ [ 1, 2, 3 ], [ 4, 5, 6 ], [ 7, 8, 9 ], [ 10, 11, 12 ] ]
 3 [[1, 2, 3],
 4 [4, 5, 6],
 5 [7, 8, 9],
 6 [ 10, 11, 12 ],
 7 [ 13, 14, 15 ] ]
   詳細は以下のとおりです。
    ■ 1行目:AI表記の getRange() で取得した範囲の値(A1:C3)を表示
    ■ 2行目:行番号・列番号指定の getRange() で取得した範囲の値(A1:C4)を表示
    ■ 3行目: getDataRange() を使ってデータが存在するセルの範囲の値(A1:C5)を表示
    5.3 セル範囲を操作する - Rangeクラス
   Rangeは、セル範囲を操作するクラスです。
                                SpreadsheetApp

☆ GAS入門 - Google スプレッドシート × +

    ← → C  

docs.google.com/spreadsheets/d/
       GAS入門 ☆ 団 ⊙
                             Spreadsheet
                                               EB - E - 1 - 10 - 17 - ...
                        Sheet
                  Range
   ここでは、Rangeクラスのメソッドについて説明します。
    メソッド
                戻り値の型
                         説明
                Range
                         範囲の値を設定
    setValue(value)
   Rangeクラスのメソッド一覧については公式サイトに記載されているため、興味のある方は参照してみてください。
   setValue(value)
   setValue(value)は範囲の値を設定するメソッドです。 setValue(value) は以下のように使用します。
   スクリプトファイル(見本)
 1 SpreadsheetApp.getActiveSheet().getRange('A1').setValue('a');
   ここでは、A1セルの値をaにします。それでは、実際にスクリプトを記述してみましょう。先ほどのシートを使いまわし
   ます。
                         2
                1
               10
                        11
                                  12
                                  15
               13
     8
   上記のシートに対して、以下のように実装します。
    1. データが存在する範囲の値を取得(値変更前)
    2. 値をコンソールログに出力
    3. A1セルの値をaにする
    4. データが存在する範囲の値を取得(値変更後)
    5. 値をコンソールログに出力
   関数名は任意ですが、ここでは myRange とします。
   GAS入門
  1 + function myRange() {
  2+ // データが存在するセルの範囲の値を取得
     let values = SpreadsheetApp.getActiveSheet().getDataRange().getValues();
  4+
     console.log(values);
  8+ // A1の範囲の値を変更
  9 + SpreadsheetApp.getActiveSheet().getRange('A1').setValue('a');
     // データが存在するセルの範囲の値を取得して表示
 11+
 12 + values = SpreadsheetApp.getActiveSheet().getDataRange().getValues();
 13 + console.log(values);
 14+}
   スクリプトを保存して「myRange」を実行すると、以下のように表示されます。
 1 [[1, 2, 3],
 2 [4, 5, 6],
 3 [7, 8, 9],
 4 [ 10, 11, 12 ],
 5 [ 13, 14, 15 ] ]
 6 [['a', 2, 3],
 7 [4, 5, 6],
 8 [7, 8, 9],
 9 [ 10, 11, 12 ],
 10 [ 13, 14, 15 ] ]
   A1セルが変更されていることがわかります。シートを表示して値が変更されていることも確認してください。
    2
                                   9
     4
                        11
                                  12
               10
     5
                                  15
               13
     6
        前半はここまで!
        休憩してリフレッシュしましょう
        後半もがんばりましょう!
    5.4 配列を使ったデータ処理
   スプレッドシートは行と列の2次元で構成されていますので、GASから2次元配列を使ってデータ処理することが一般的で
   す。
   ここでは、[ ['a', 'b', 'c'], ['d', 'e', 'f'] ]という2次元配列を「配列練習用」シートに配列の値を書き込むスクリプトを実装して
   みます。
   「配列練習用」シートは事前に作成しておいてください。関数名は任意ですが、ここでは myArray とします。
   GAS入門
  1 + function myArray() {
     let sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('配列練習用');
  5+ // 配列を作成
  6 + let data = [
  7+ ['a', 'b', 'c'],
      ['d', 'e', 'f']
  9+];
 10 +
 11 + // シートに配列データを書き込む
 12 + sheet.getRange(1, 1, data.length, data[0].length).setValues(data);
 13 + }
 14
   2行3列のデータ範囲を取得したいので、 getRange の第3引数には data.length 、第4引数には data[0].length を指定してい
   ます。
   2次元配列であるdataに対して、lengthプロパティを指定すると要素数を取得できます。この場合、 data.length は['a', 'b',
   'c']と['d', 'e', 'f']で2です。
   data[0]は['a', 'b', 'c']なので、 data[0].length は、'a'と'b'と'c'で3になります。
   スクリプトを保存して「myArray」を実行すると、2次元配列の値がA1セルから設定されていることがわかります。
    2
    5.5 カスタム関数
   スプレッドシートでは、式を利用できます。例えば、あるセルに「=SUM(A1:C5)」という式を入力すると、A1からC5の
   範囲の数値を合計した数値が表示されます。
   SUM の部分を関数といいます。
   この SUM のような関数は最初から用意されているのですが、ユーザが独自に関数を定義することもできます。これを、カ
   スタム関数といいます。
   例えば、スクリプトエディタで以下のような関数を作成したとします。
   GAS入門
  1 + function myCustomFunc(data){
  2 + return data * 2;
  3+ }
   スクリプトを保存してスプレッドシートに戻り、任意のセルに「=myCustomFunc(1)」と入力してみてください。該当セ
   ルには2と表示されます。
   myCustomFunc は引数(この場合1)を2倍にした値をreturnしているので、このように表示されます。
    A1
                                  =myCustomFunc(1)
                                                              C
      3
   「=myCustomFunc(A1)」のように、A1形式で引数を指定することも可能です。
    B1
                                   =myCustomFunc(A1)
                                                              C
   本章の学習は以上です。お疲れさまでした。
    まとめ
   本章では以下の内容を学びました。
    ■ Sheetはシートを操作するクラスで、代表的なメソッドとして以下のメソッドがある
    メソッド
                                   戻り値の型
                                            説明
                                            A1表記でセルの範囲を取得
    getRange(a1Notation)
                                   Range
                                            行番号・列番号指定でセルの範囲を取得
    getRange(row, column, numRows, numColumns)
                                   Range
    getDataRange()
                                   Range
                                             データが存在するセルの範囲を取得
    ■ Rangeはセル範囲を操作するクラスで、代表的なメソッドとして以下のメソッドがある
                戻り値の型 説明
    メソッド
    setValue(value)
                Range
                         範囲の値を設定
   前章と本章で、Spreadsheetサービスに関する基礎知識をすべて学習しました。
   次章では応用編として、スプレッドシートに対する自動化を実践します。教材に沿って実践することで、自動化の手順が
   身につけられるので、頑張って習得しましょう。
                     理解度を選択して次に進みましょう
                    ボタンを押していただくと次の章に進むことができます
                                                    80~100%
                      最後に確認テストを行いましょう
                         下のボタンを押すとテストが始まります。
                                              テストをはじめる
                     教材をみなおす
                                  5/13ページ
   前に戻る
                                                                      次に進む
く一覧に戻る
                                       ■ 改善点のご指摘、誤字脱字、その他ご要望はこちらからご連絡ください。
                                       © SAMURALInc. 利用規約 プライバシーポリシー 運営会社
```

SAMURAI TERAKOYA

① タイムライン

☆ ホーム

田 教材

□ 課題

Q Q&A

ク 学習ログ

の レッスン

よくある質問

リリースノート

教材

Q 検索

○40分 ~ ○ 理解

5.1 本章の目的

ラス」について学習しました。

Sheetはシートを操作するクラスです。

T GAS入門 - Google スプレッドシート × +

ファイル 編集 表示 挿入 表示形式 データ ツー

本章では、以下を目標にして学習します。

■ シートを操作する「Sheetクラス」の使用方法を理解する

使ったデータ処理やカスタム関数についても解説します。

5.2 シートを操作する - Sheetクラス

■ セル範囲を操作する「Rangeクラス」の使用方法を理解する

クラス・Rangeクラス

ホーム > 教材 > Google App Scriptで業務自動化プログラムを作ろう > スプレッドシートを操作しよう #2 Sheetクラス・Rangeクラス

5章 スプレッドシートを操作しよう #2 Sheet

シートを操作する「Sheetクラス」の使用方法、セル範囲を操作する「Rangeクラス」の使用方法を学びま

前章では、アプリケーション自体を操作する「SpreadsheetAppクラス」、スプレッドシートを操作する「Spreadsheetク

本章では、シートを操作する「Sheetクラス」、セル範囲を操作する「Rangeクラス」について解説します。また、配列を

Spreadsheet

Sheet

Range

SpreadsheetApp

В

1 GAS (Google App Script) の概

2 スクリプトエディタとダッシュボ

ードの使い方を理解しよう

3 GASの制限事項を理解しよう

Spreadsheetクラス

5.1 本章の目的

4 スプレッドシートを操作しよう

5 スプレッドシートを操作しよう

#2 Sheetクラス・Rangeクラス

5.2 シートを操作する - Sheetクラ

#1 SpreadsheetAppクラス・

要を理解しよう

本文 目次 質問一覧 1件