2. スクリプト作成の基本操作

ここでは、スクリプト作成に関する基本操作方法について説明します。

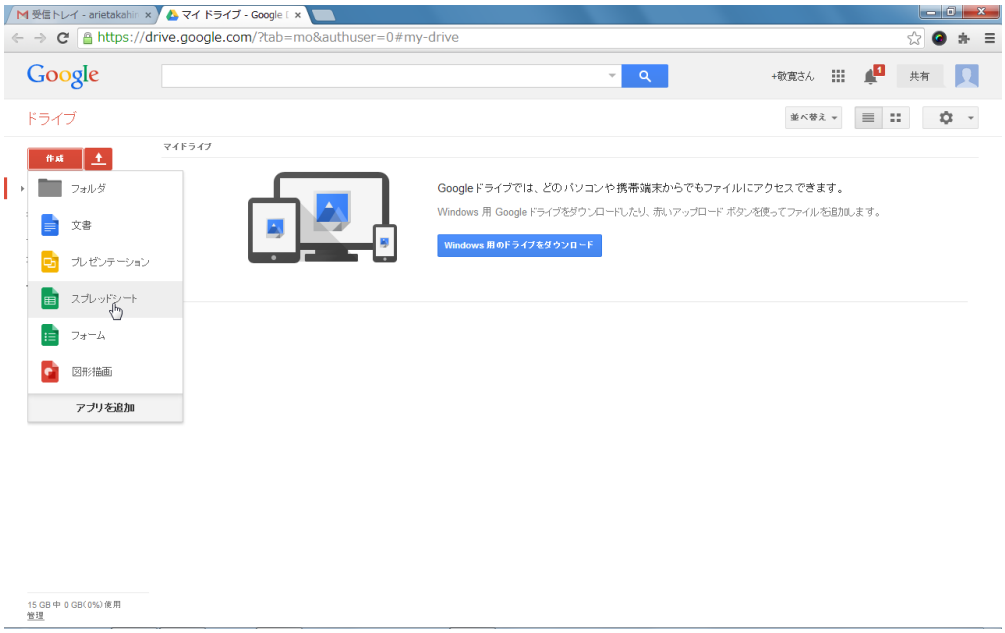
2.1. スクリプトエディタの起動

Google Apps Script のスクリプトは「スクリプトエディタ」と呼ばれる専用のツールを使って入力します。まずはスクリプトエディタを起動する事から始めましょう。

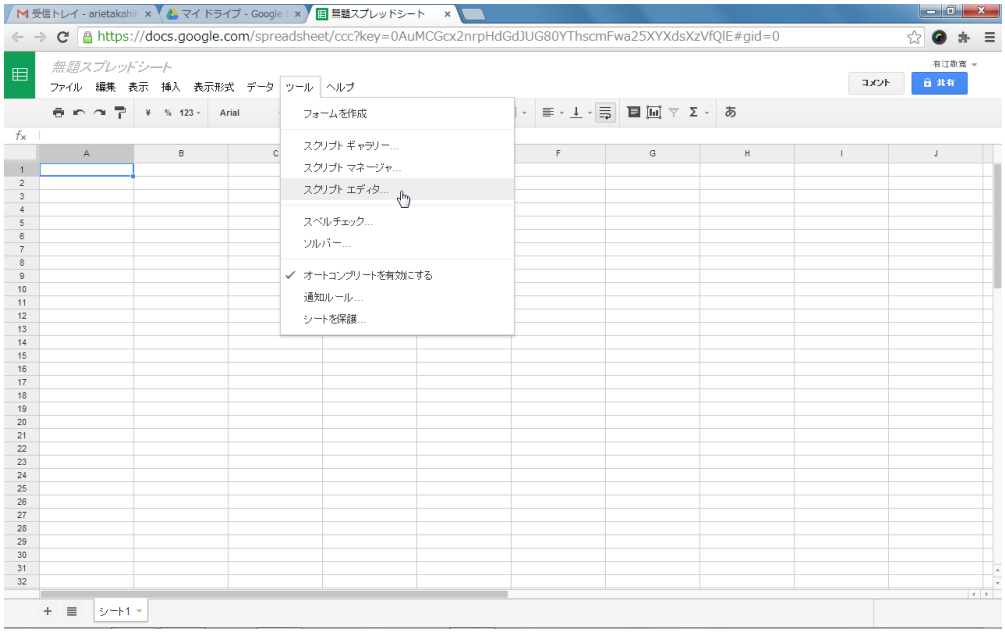
Gmail にログインします。

「アプリ」メニューから「ドライブ」を選択します。

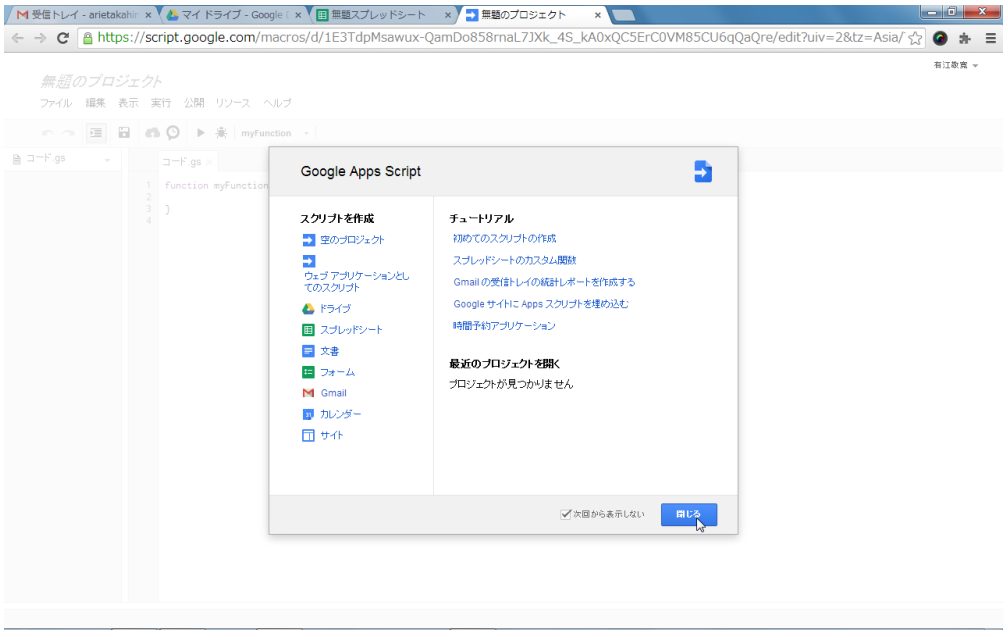
Google ドライブが開いたら「作成」メニューから「スプレッドシート」を選択します。



新規のスプレッドシートが開いたら「ツール」メニューから「スクリプト エディタ...」を選択します。

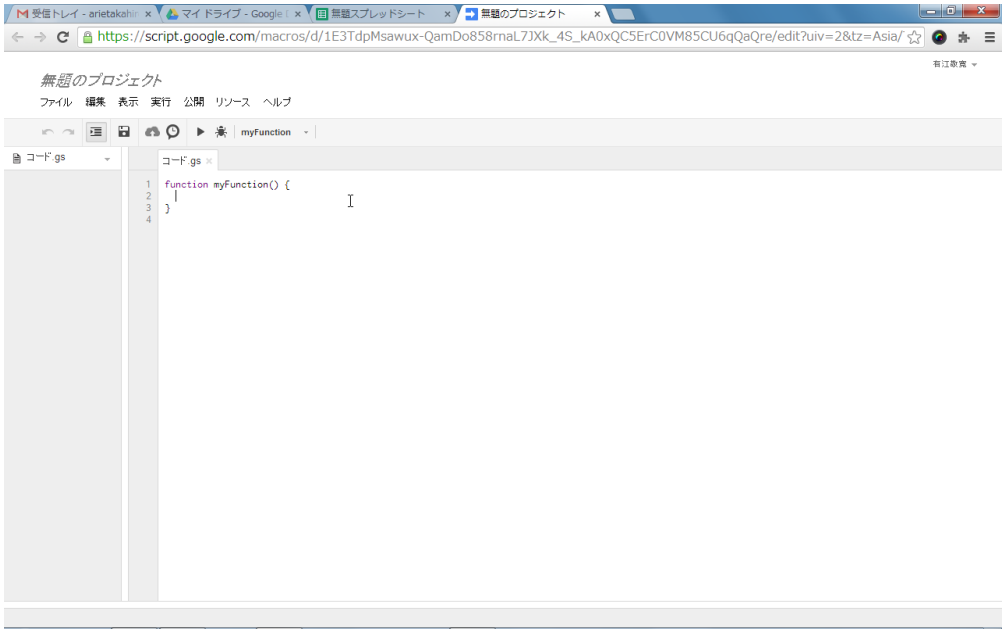


チュートリアル画面が表示された場合は「閉じる」を押しましょう。



スクリプトエディタが起動し、以下の 3 行が表示されます。

```
01:function myFunction() {
02:
03:}
```

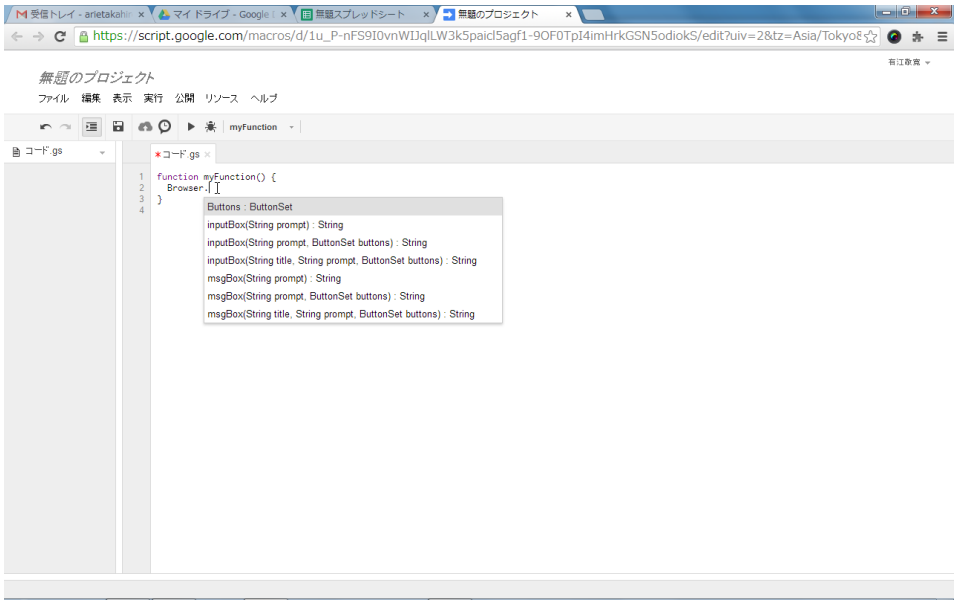


これにて、スクリプトエディタの起動は完了です。

2.2. スクリプトの作成と実行

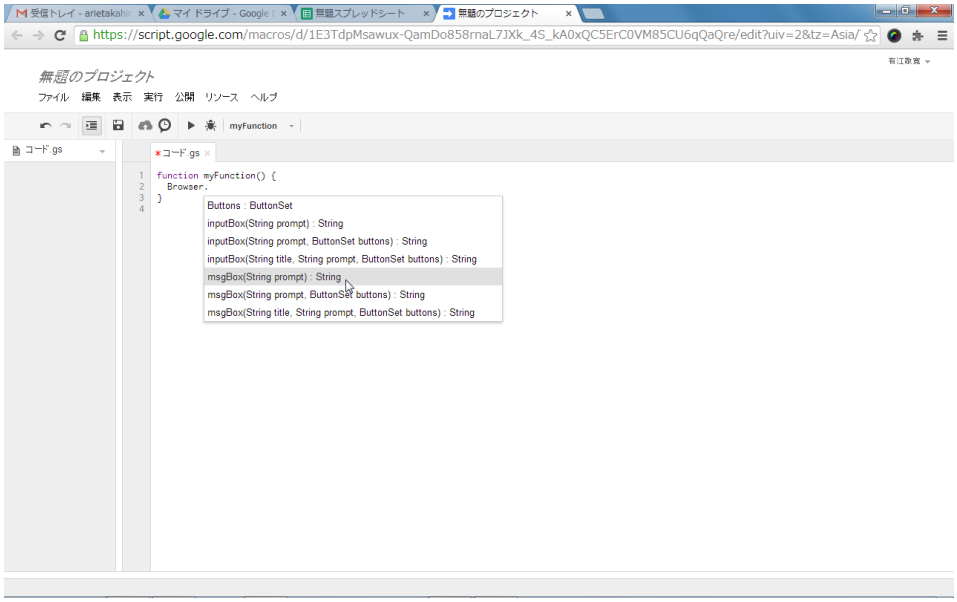
ここでは、前節で説明したスクリプトエディタを使ってメッセージボックスを表示するだけの簡単なスクリプトを作成し、実際に実行してみましょう。今までプログラミングに全く携わっていなかった方には意味不明な個所はあるかとは思いますが、とりあえず細かい事は置いておき、スクリプトを作って動かしてみる経験をする事を目的としてやってみましょう。

まず、空白行である 2 行目に「Browser.」（注：B のみ大文字、かつ、r の後のピリオドまで入力する必要あり）と入力すると、スクリプトエディタの入力支援機能が表示されます。これは、Google Apps Script で定義されているキーワードを途中まで入力すると、全部タイピングしなくて済むように、その後に続くキーワードを選択できるようにしてくれる機能です。Microsoft 製品で言うところの「インテリセンス」ですが、Google 製品では特別な呼び方はなく「入力支援機能」と呼ばれています。大変、重宝する機能なのですが、たまに動きが遅くなるのと、携帯の漢字予測変換ように頻繁に使う順に並び換ったりはしませんので、その辺は寛大な心で使わなくてはなりません。

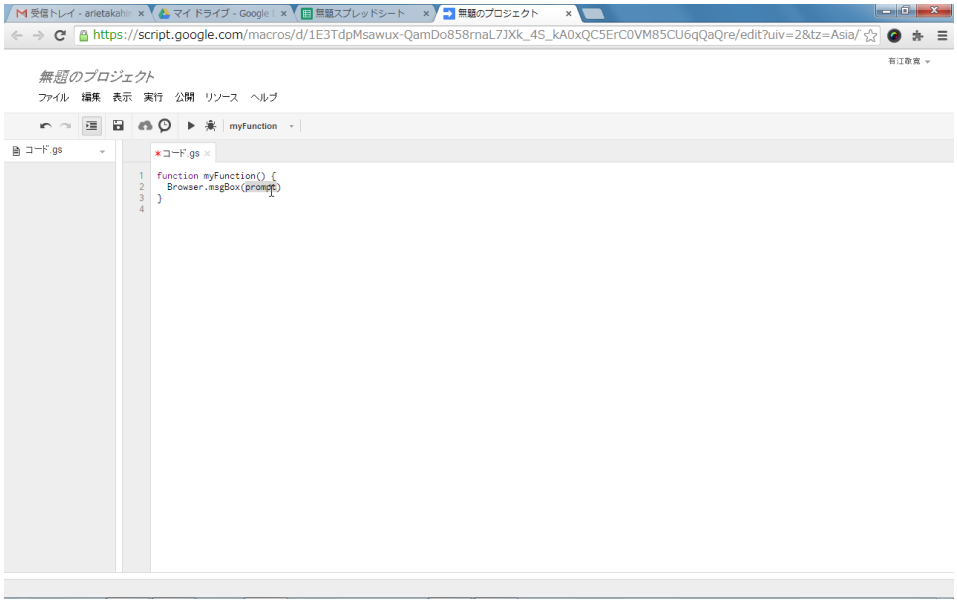


また、入力支援機能を使わず全てタイピングする場合は、大文字と小文字をしっかり区別して入力する必要があります。msgBox が正しい入力ですが、MsgBox や msgbox は誤りです。

選択肢の中から「msgBox(String prompt): String」を選択します。



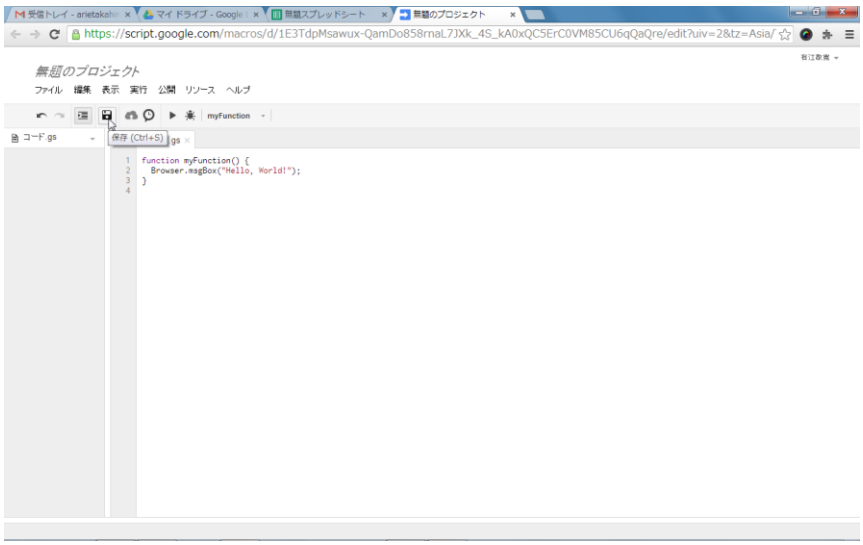
「prompt」と表示されている部分をクリックします。



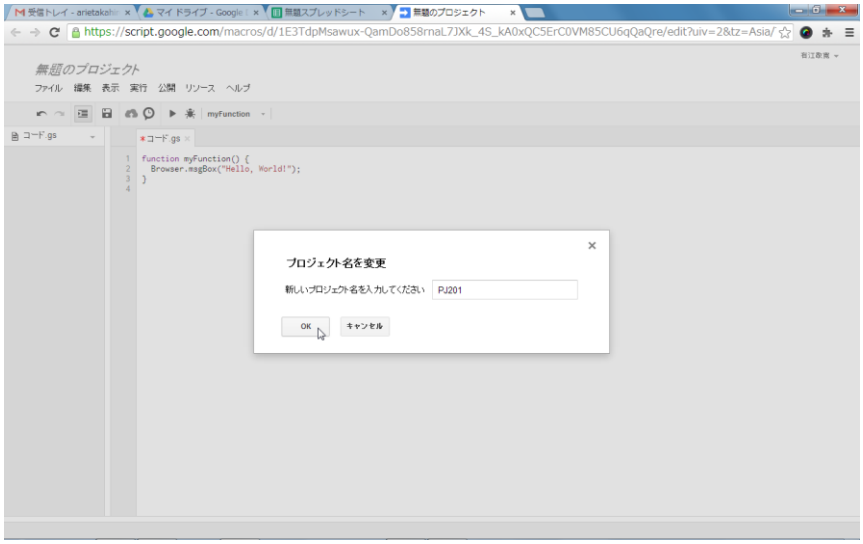
「prompt」の文字を消して 「"Hello, World!"」と入力します。ダブルクォーテーション「 “」の間に挟んだ文字であれば何でもよいのですが、最初のサンプルプログラムは「Hello, World!」と表示するのが定番となっています。なぜ定番なのか興味のある方は章末のコラムをご覧下さい。また、文の最後にセミコロン「;」を付けるのを忘れないようにしましょう。Google Apps Script では、通常、文末にセミコロンを入れます。細かい事を言うと、文法的には入れなくても大丈夫なのですが、文末にセミコロンを入れておかないと、スクリプトが複数行になった時に入力支援機能が正しく動作しません。今回の例では、結局、以下のように入力されていれば OK です。

```
function myFunction() {
  Browser.msgBox("Hello, World!");
}
```

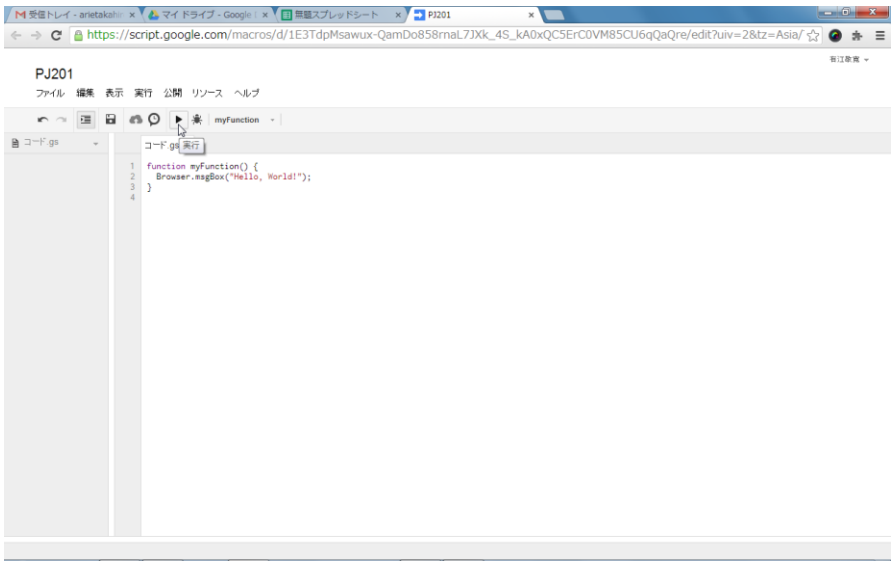
スクリプトを入力できたら、フロッピーディスクのマークの「保存」ボタンを押してスクリプトを保存します。



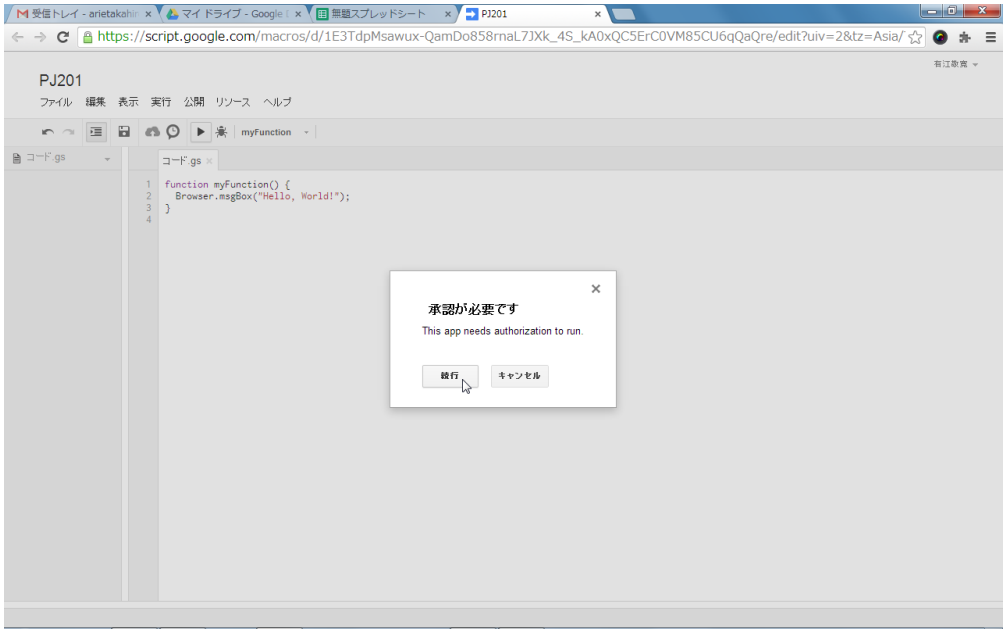
「Rename Project（プロジェクト名を変更）」ダイアログボックスが表示されたら、プロジェクト名を入力します。ご自分の好きな名前で構いません。しかしながら、好きな名前と言われても、とりあえず本に書いてある通りに試しているだけなので好きな名前も何もない、という声もよく耳にします。そういう場合は、とりあえず次の方法を参考にしてみて下さい。それは、とりあえず 2～3 文字の略称と数桁の数字を合わせて名前を付けておく方法です。例えば、プロジェクト名であれば、Project の略称で「PJ」、第 2 章の最初のプロジェクトなので数字は「201」、という事で「PJ201」という名前を入力して「OK」ボタンを押します。（数字の部分は「日付＋連番」でもいいですね。）



前述の操作でつけたプロジェクト名は右上に表示されます。続いて、三角マークの「実行」ボタンを押してスクリプトを実行します。

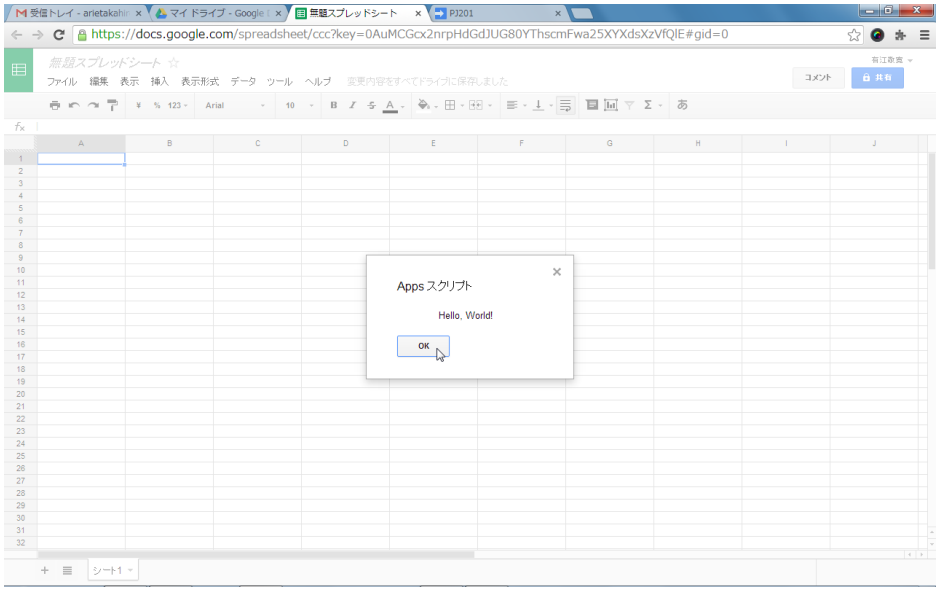


すると「承認が必要です」画面が表示されますので「続行」ボタンを押します。

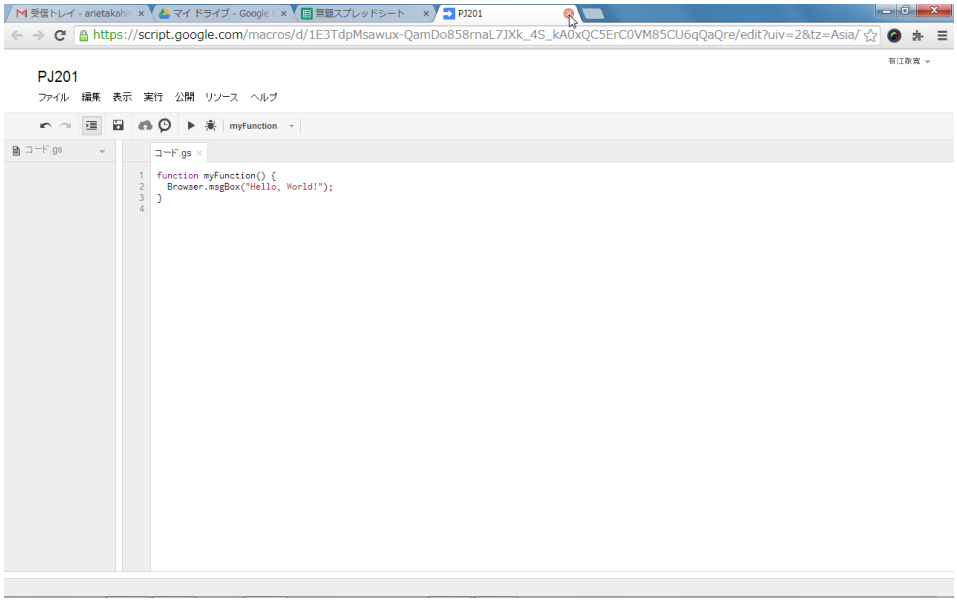


スクリプトの実行を許可するため「承認する」ボタンを押します。

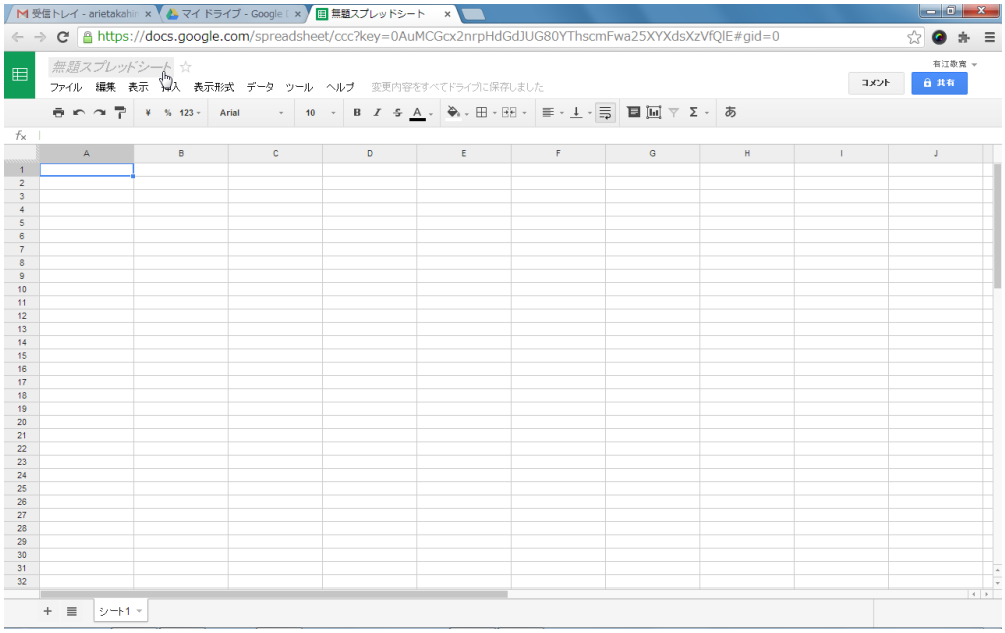
ここで「実行ボタンを押したのに、何も起こらない」と思った方が多いはずです。実は、スクリプトを実行しても、スクリプトエディタの画面では何も起こりません。なぜならスクリプトはスクリプトエディタの画面ではなく、スプレッドシートの画面で実行されるからです。よって、実行結果を確認するために表示画面をスプレッドシートに切り替えます。すると、メッセージボックスが表示されているはずです。ここで「OK」ボタンを押せばスクリプトは終了します。



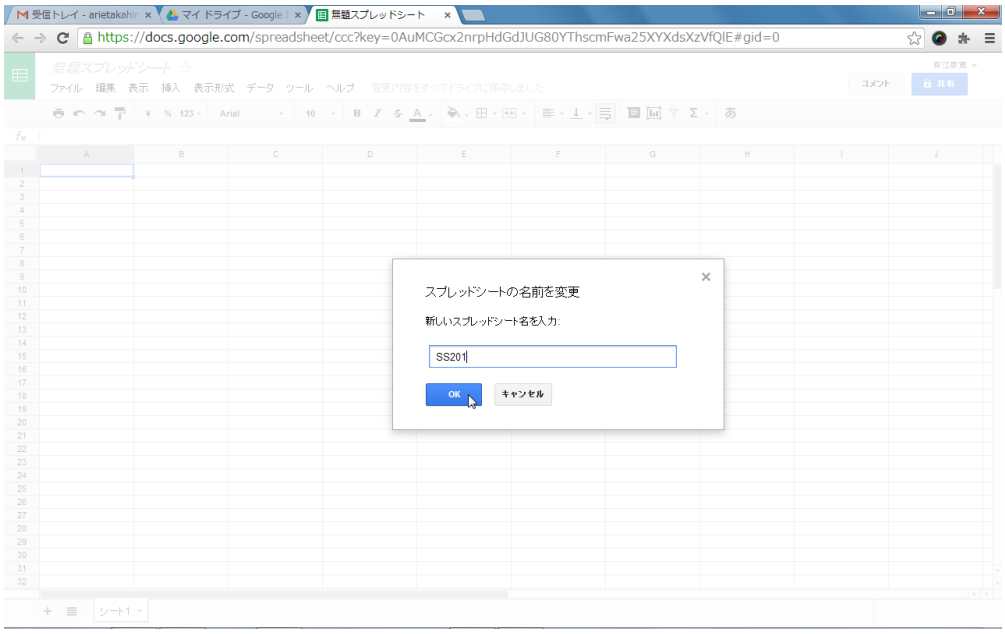
スクリプトエディタの画面に戻り、タブの×ボタンを押して、スクリプトエディタを閉じます。



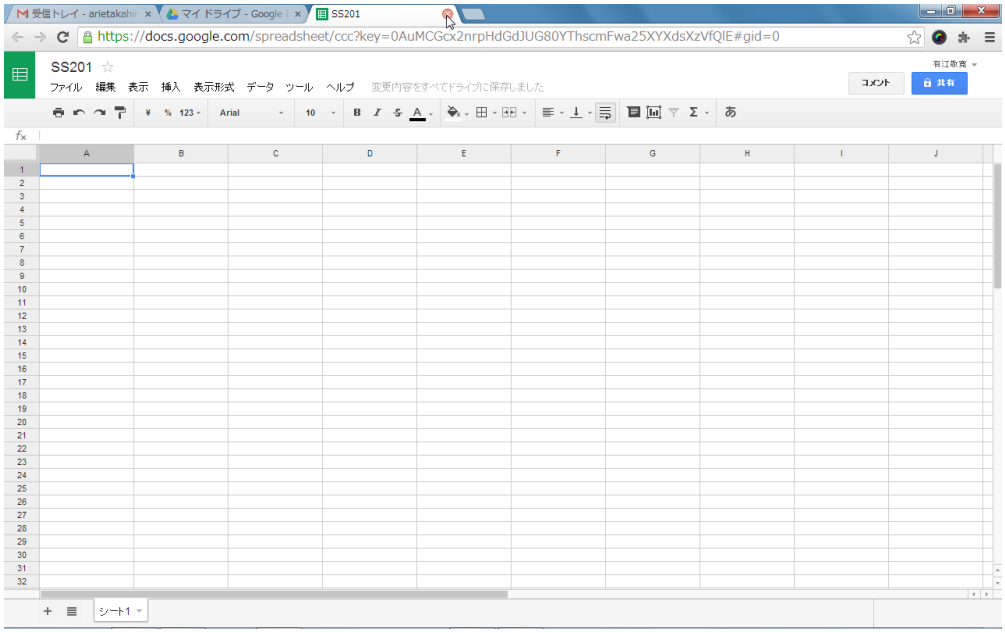
続いて、スプレッドシートを保存するわけですが、Google ドライブでは「無題スプレッドシート」という名前でスプレッドシートが自動的に保存されています。これは、何らかの理由で急にネット接続が切れた場合でも、編集した内容が全部無くならないようにするための仕組みです。自動保存された「無題スプレッドシート」という名前を変更したい場合は、スプレッドシート名が表示されている部分をクリックします。



「スプレッドシートの名前を変更」ダイアログボックスが表示されますので、スプレッドシート名を入力して「OK」ボタンを押して下さい。これも好きな名前で構いません。例えば、スプレッドシートであれば、SpreadSheet の略で「SS」、第 2 章の最初のスプレッドシートなので数字は「201」、という事で「SS201」という名前を入力して「OK」ボタンを押します



最後に、タブの×ボタンを押して、スプレッドシートを閉じます。



保存されたスプレッドシートが Google ドライブの一覧に表示されます。

これにて、スクリプトの作成と実行は完了です。

Google ドライブの紹介が表示されている場合は、×ボタンを押して閉じておきましょう。

【スクリプトの解説】
Google Apps Script で記述したスクリプトは「関数」と呼ばれる単位で実行します。よって、通常は、スクリプトで実行させたい処理が「関数」という単位でまとまっています。関数は以下のような書き方で定義します。

```
function 関数名() {
  処理
}
```

関数には基本的に好きな名前をつける事ができますが、実は、用途によって既に決められている名前もあるので、それと重複しないようにするには、関数名の頭に「my」を付けておくのが無難です。スクリプトエディタを起動した時の関数名も頭に「my」が付いていますね。

```
function myFunction() {
  Browser.msgBox("Hello, World!");
}
```

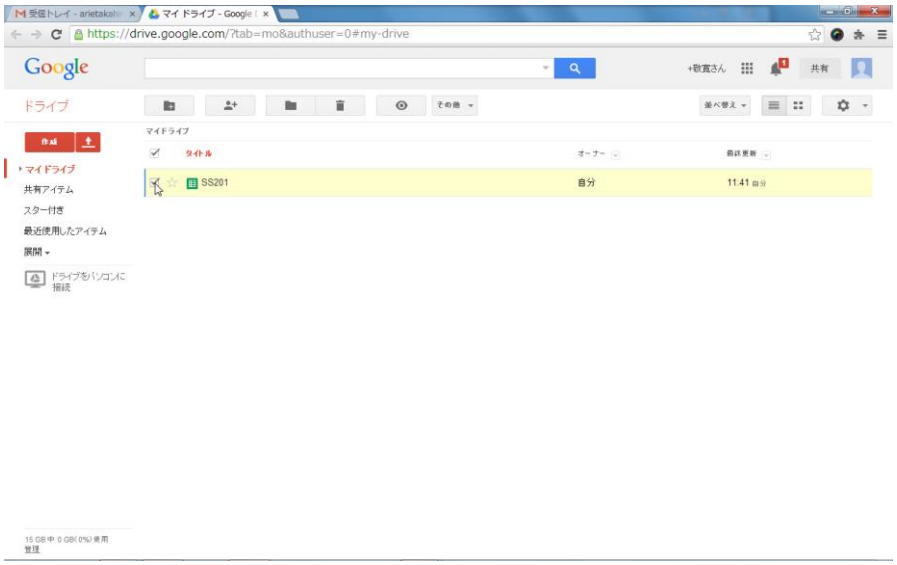
今回のスクリプトでは、メッセージボックスを表示する処理を行う「Browser.msgBox()」を使っています。これは、括弧(かっこ)の中に記述された文字列をメッセージボックスとして表示するだけの簡単な処理ですが、時間がかかるスクリプトの最後の行に記述しておけば、スクリプトが最後まで行われたのが明確に分かるので、大変重宝します。

【参考にしたサイト】
Google Apps Script > Tutorial: Creating Your First Spreadsheet Script
http://code.google.com/intl/ja/googleapps/appsscript/articles/yourfirstscript.html

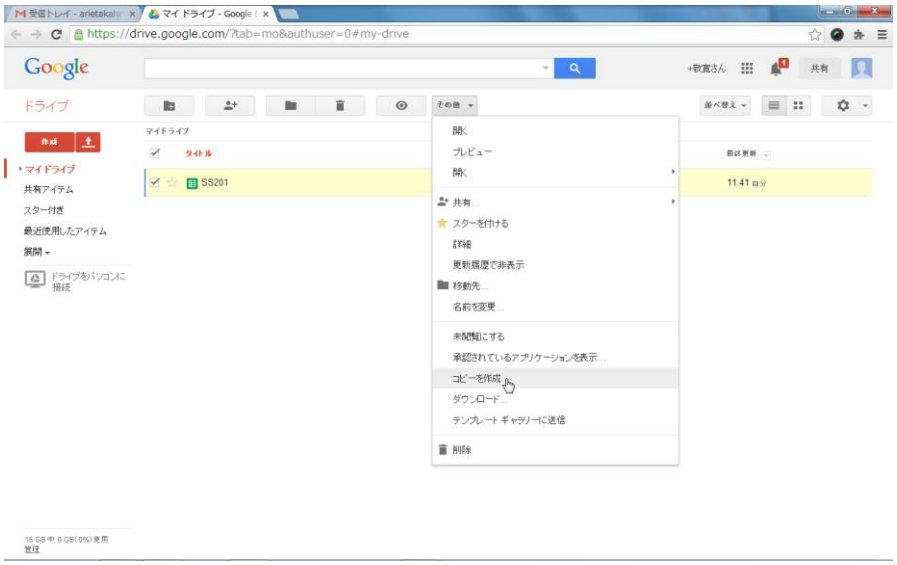
2.3. スプレッドシートのコピーと削除

同じようなスクリプトを何個も作成する場合や、作ったスクリプトのバックアップを残しておきたい場合は、スプレッドシートごとコピーしておくのが便利です。ここでは、その手順について説明します。

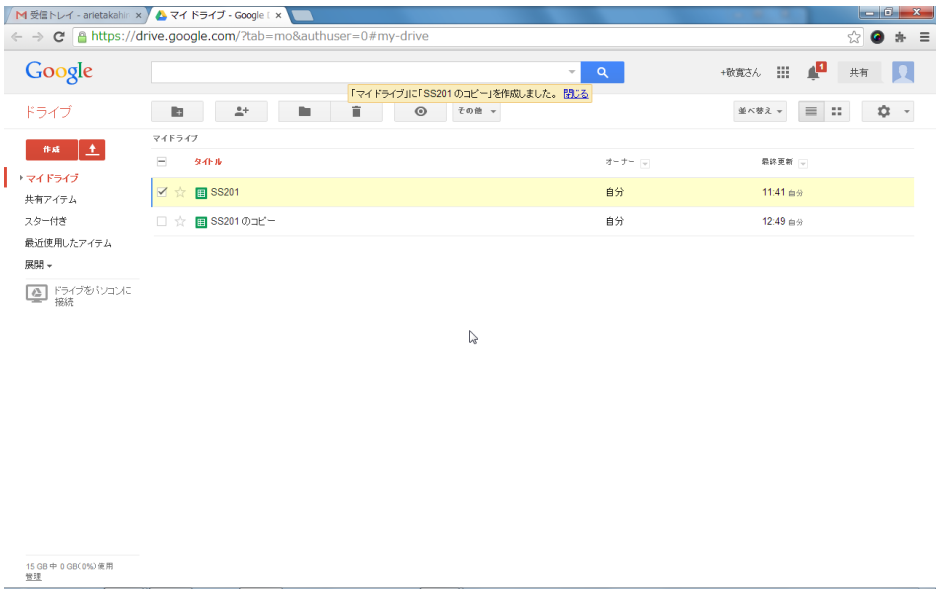
まず、Google ドライブの一覧画面で、コピー対象のスプレッドシートをチェックします。



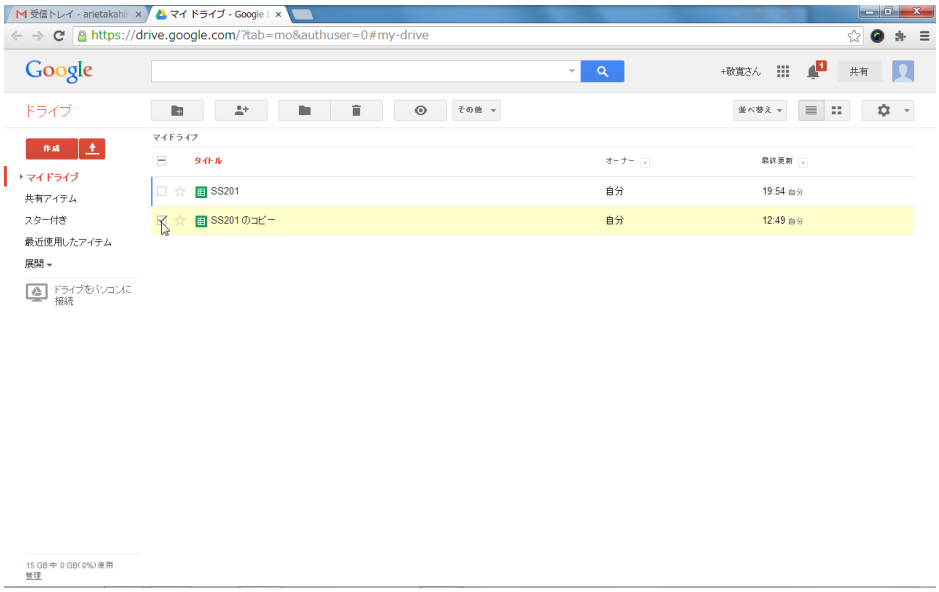
「その他」メニューから「コピーを作成」を選択します。



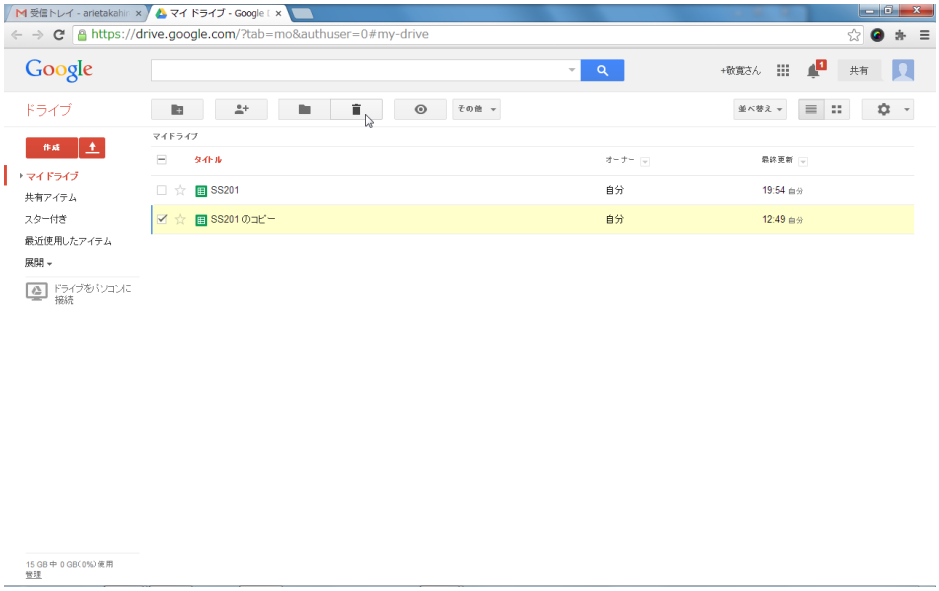
Google ドライブの一覧に、コピーして作成したスプレッドシートが表示されます。表示されない場合は「F5」キーを押してページを再読み込みしてみましょう。



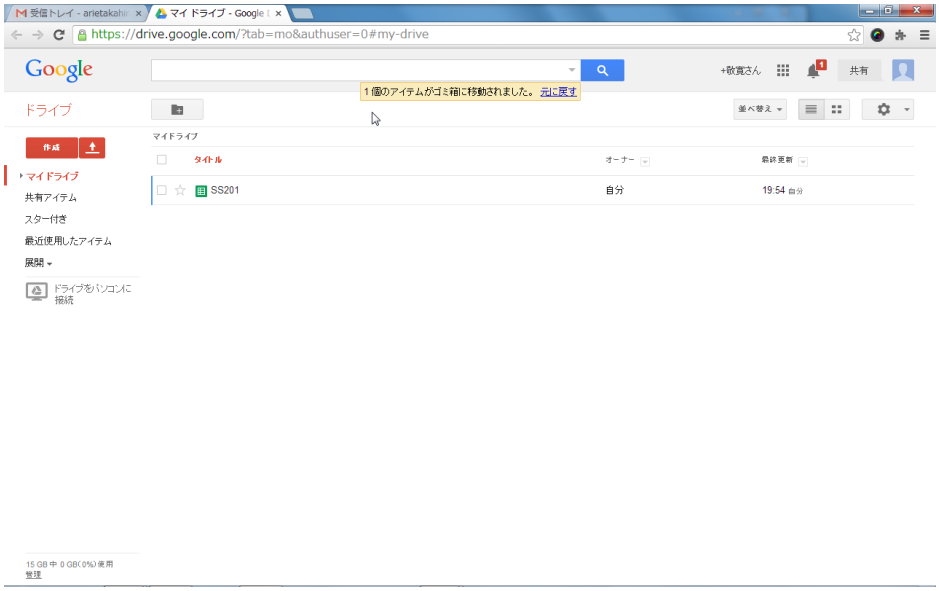
コピーの方法が分かったら、削除の仕方も覚えておきましょう。削除方法はとても簡単で、まず、削除したいスプレッドシートをチェックします。



ドライブの一覧の上に表示された「削除」ボタンをクリックします。



削除したスプレッドシートが一覧に表示されなくなります。



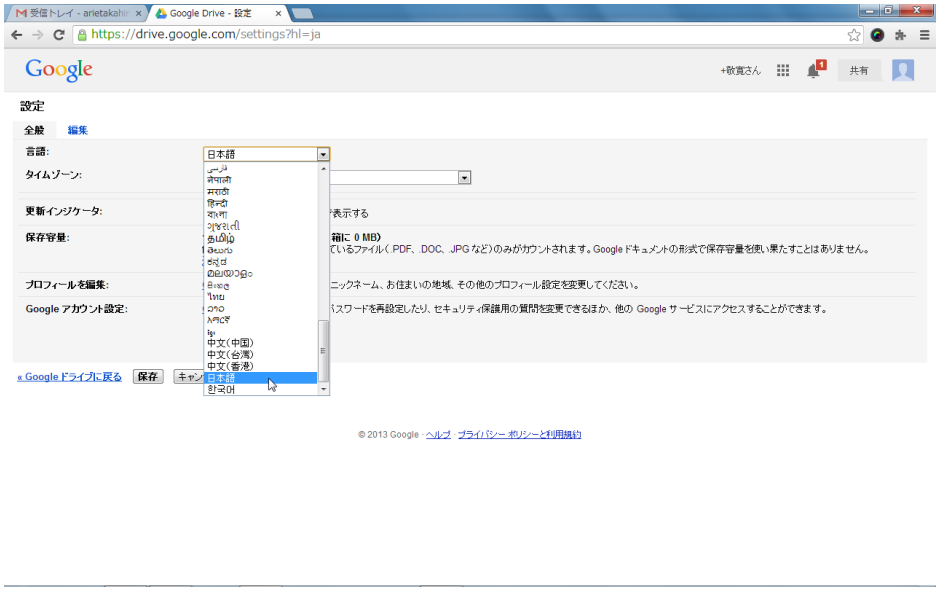
これにて、スプレッドシートのコピーと削除は完了です。

【コラム】Google ドライブのタイムゾーン設定

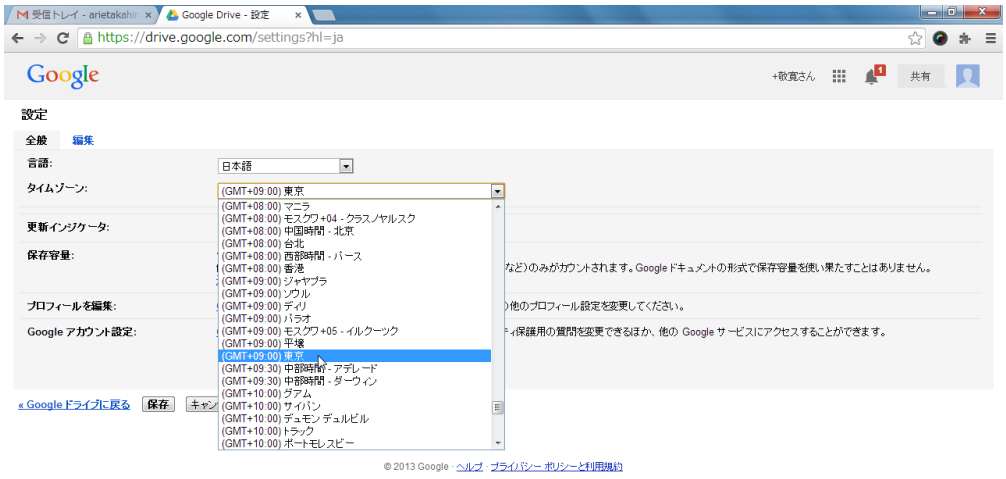
Google ドライブを初めて使った方は、言語とタイムゾーンの設定をしておきましょう。

右上の歯車のマークを押した時に表示されるメニューから「設定」を選択します。

「言語：」に「日本語」を選択します。



「タイムゾーン：」に「(GMT+09:00)東京」を選択します。



最後に「保存」ボタンを押します。

【コラム】最初のプログラムは何故「Hello, World!」 なのか？

プログラミングの勉強をした経験がある方であれば既にご存知かもしれませんが「Hello, World!」という文字列を表示するプログラムは、入門書の最初のサンプルプログラムとなっている事がよくあります。別に「Hello, World!」以外の文字列でも構わないはずなのですが、なぜかこの文字列が愛されています。

その起源は、ブライアン・カーニハン（Brian W. Kernighan）とデニス・リッチー（Dennis M. Ritchie）による著書「プログラミング言語 C」(原題 : The C Programming Language)であると言われています。同書は、初版が 1978 年に出版（日本語訳は 1981 年に出版）され、当時、「C 言語」というプログラミング言語を学ぶ人達のバイブルとして重宝されました。

この本の中で『プログラミング言語を学ぶ最善の方法は、実際にプログラムを書くことであり、最初に書くべきプログラムはどんな言語も同じである。例えば「hello, world という単語を印字せよ」というものだ。』という教えが書かれていました。

実際に C 言語で記述してみると次のようになります。

```
#include <stdio.h>
main()
{
  printf("hello, world¥n");
}
```

つまり、このバイブル本に書かれていた最初のサンプルプログラムが今でもオマージュされ、色々な入門書の最初のサンプルプログラムとなっているのです。

しかしながら、よく見てみると“hello, world”は全て小文字ですし、ビックリマークも付いていません。したがって、本当の通であれば、“hello, world”とすべきなのでしょうが、そこまでコダワリを前面に出すと、大抵の場合はドン引きされるので、ちょっと見栄えを良くして”Hello, World!”とするぐらいがちょうどよいのではないかと思っています。