WebClapModoki_GAS

（google スプレッドシート＆Google Apps Script（GAS）で Web拍手っぽい何かを実装してみた）

処理内容・導入手順の詳細説明スライド

2022/12/25 初版

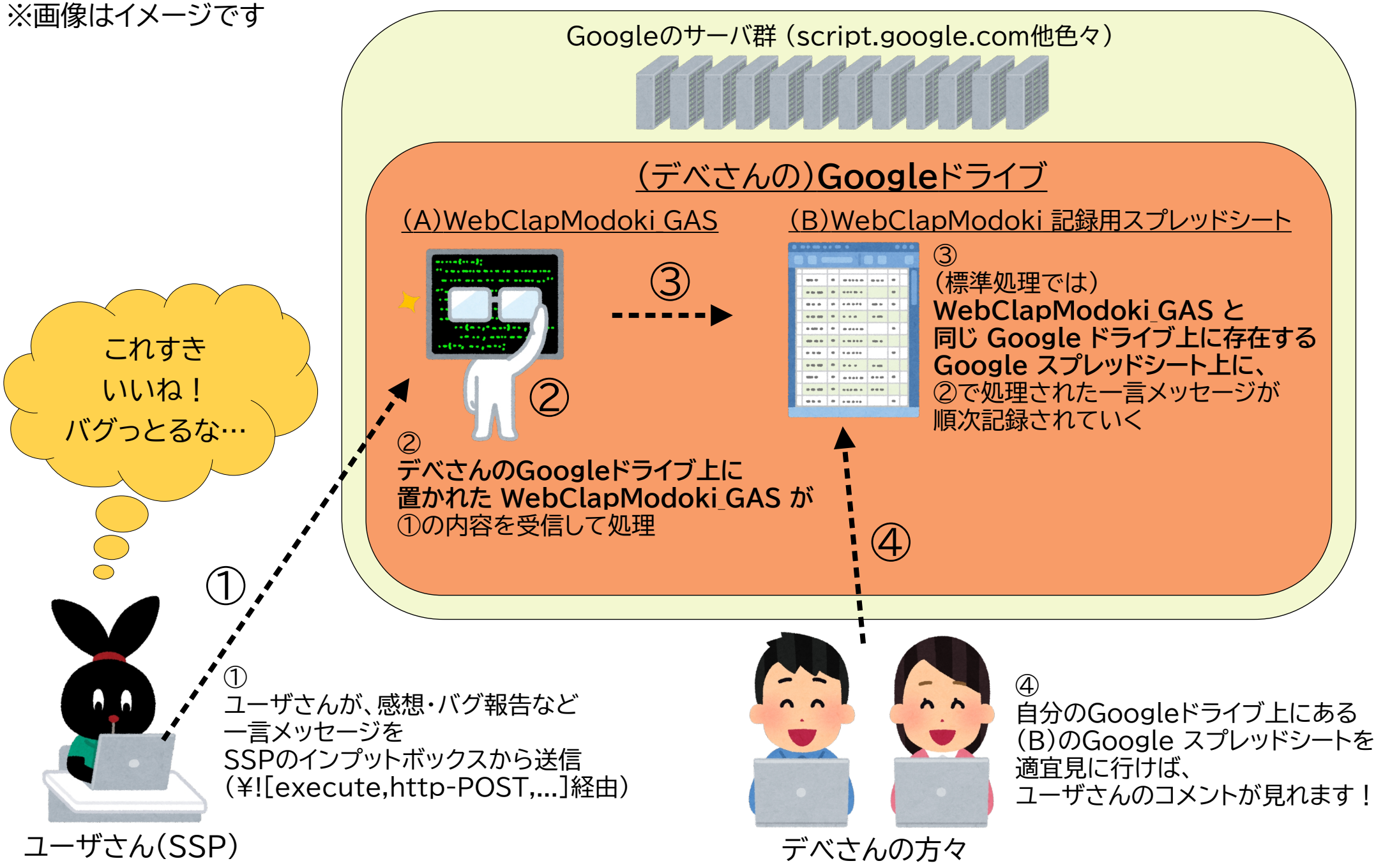
文責：dullNeko

GitHub: https://github.com/dullNeko/WebClapModoki_GAS
Twitter: @dullNeko
Mastodon: @dullNeko@ukadon.shillest.net

●目次

(1)【処理の図解】これ、どんな処理するの？

※画像はイメージです



（2）【準備するもの】必要なものは何？

以下の【1】【2】【3】【4】が必須です。

【2】【3】【4】については、
適切な場所への設置・設定書き換え等を行う必要があります。

【1】　Google アカウント
＋Google ドライブの空き容量（数MBくらい）

【2】　dullNeko の Google ドライブ上で公開されている WebClapModoki (template) のコピー
https://docs.google.com/spreadsheets/d/1j_h4weHihW-uVeTFtHTwrs6QwRPAT36sEBf3DPbo3ms/edit#gid=0

【3】　dullNeko の GitHub リポジトリにある WebClapModoki_GAS.txt の中身
https://raw.githubusercontent.com/dullNeko/WebClapModoki-GAS/main/WebClapModoki-GAS.txt

【4】　デベさんのゴーストさん内での \![execute,http-POST,...] の記述

※　POSTメソッドを使ってHTTPリクエストを送れるモノなら、
送信側はSSPに限りません：
例えばご自分のWebサイトに設置したスクリプトとか、
あるいは自作ツールに実装したバグ報告機能とか、もOK（なはず）です。

（3）【環境作成・設定方法】どうすれば動くの？

大きく分けて、次の4段階で設定していきます。

（3.1）Google ドライブ上に必要なファイルを用意
（3.2）Google Apps Script の準備／動作確認／設定変更
（3.3）Google Apps Script のデプロイ／WepアプリURL取得
（3.4）SSP側でのHTTPリクエストの実装／動作テスト

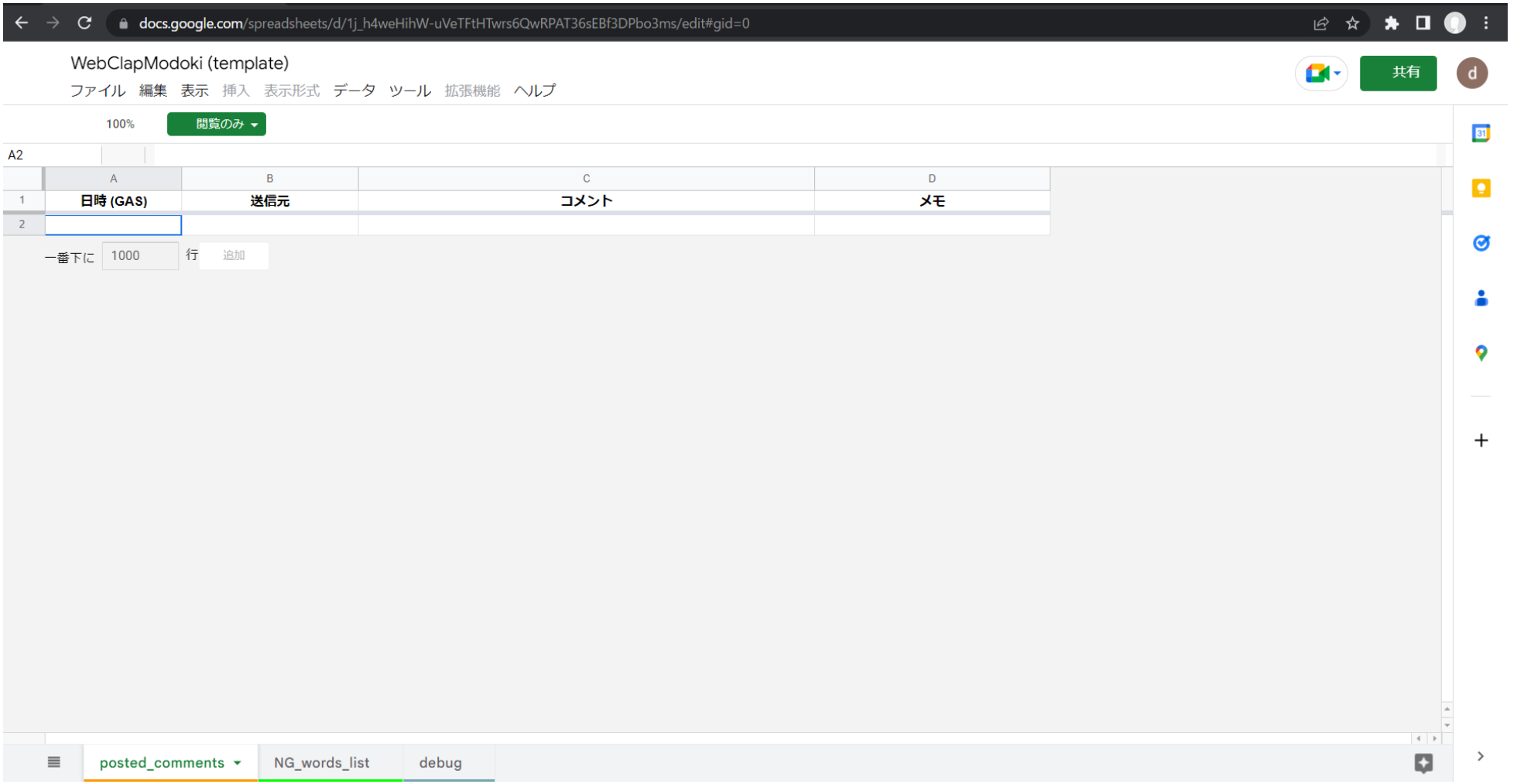
次スライドから、順番に説明していきますね。

（3.1） Google ドライブ上に必要なファイルを用意 （1／23）

（1）まず、**Googleアカウントでログインした状態のブラウザ上**で
dullNeko の Google ドライブ上で公開されている、次のスプレッドシートを開いて下さい。

https://docs.google.com/spreadsheets/d/1j_h4weHihW-uVeTFtHTwrs6QwRPAT36sEBf3DPbo3ms/edit?usp=sharing

…次の画面が出てきたでしょうか？



（3.1）Google ドライブ上に必要なファイルを用意（2／23）

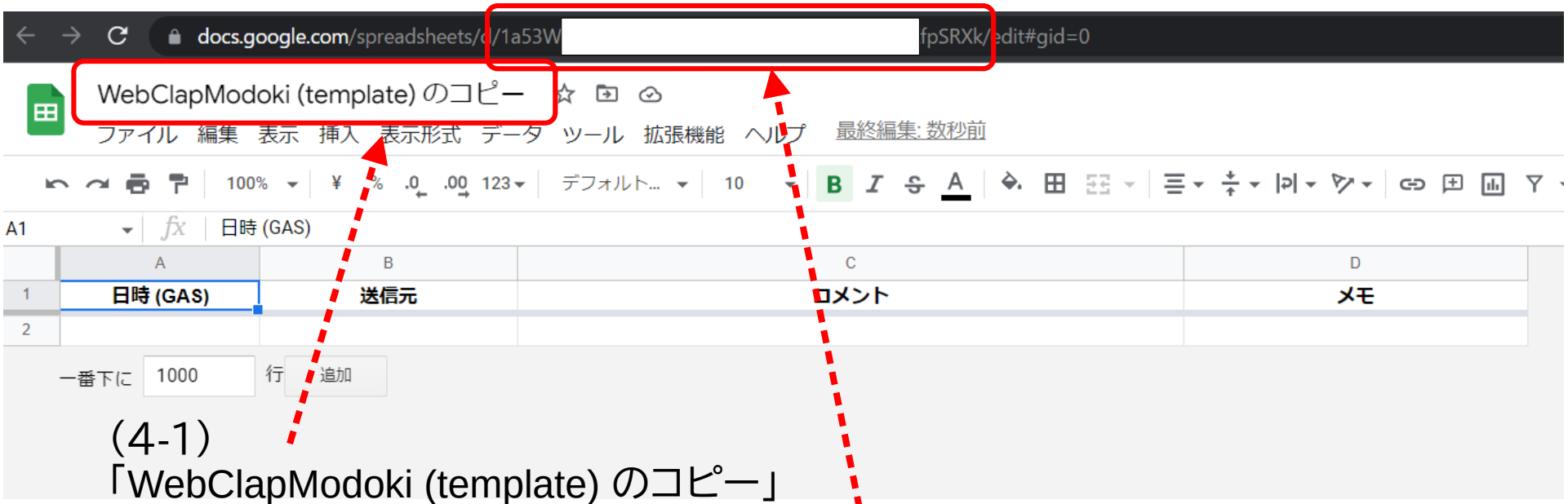
（2）
メニューにある
「ファイル」→「コピーを作成」
を開きます。

※　「コピーを作成」が
グレーアウトして選択できない場合は、
「Googleアカウントのログイン」を
やり直してみて下さい。

（3.1） Google ドライブ上に必要なファイルを用意（3／23）

（3）
「コピーを作成」をクリックして実行します。

（3.1）Google ドライブ上に必要なファイルを用意（4／23）



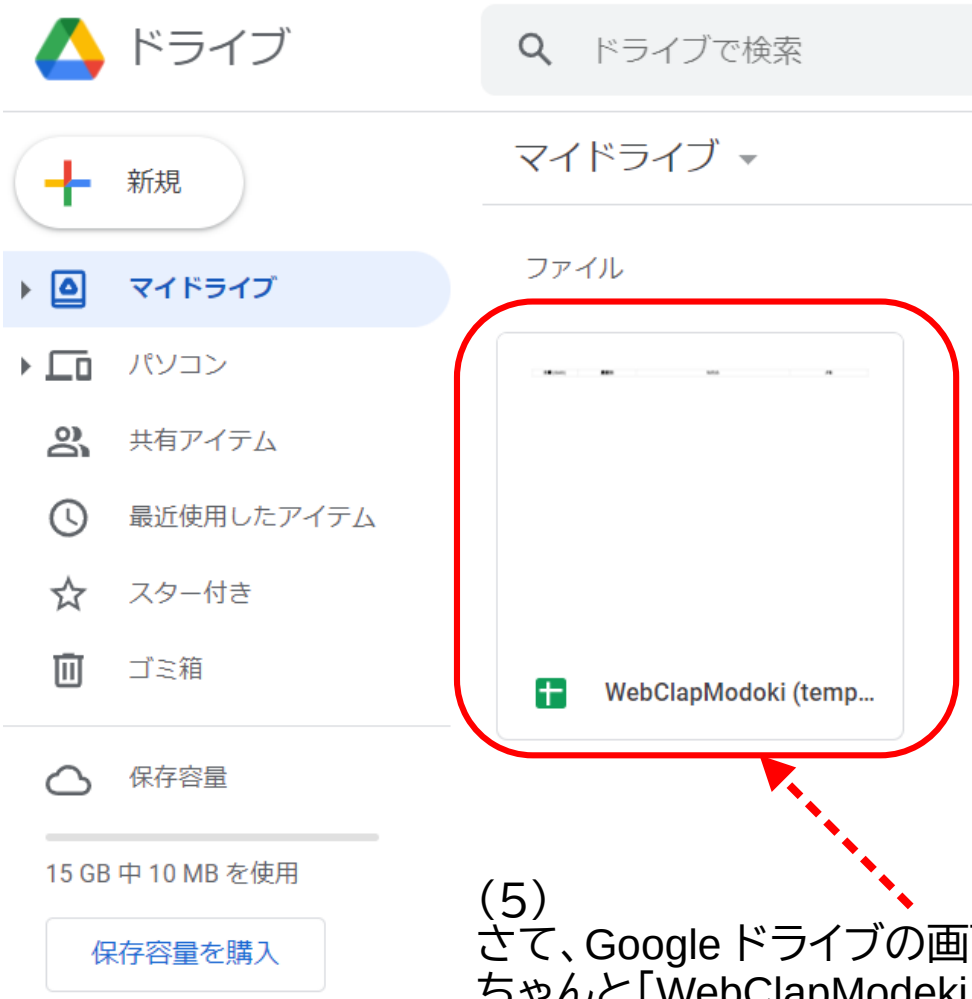
(4-1)
「WebClapModoki (template) のコピー」
が作成されればOK！

(4-2)
※重要ポイント※
ここの白抜きで一部隠してある部分が
「（スプレッドシートの）id 」と呼ばれる、
「スプレッドシートを特定する」大変重要な情報です。

https://docs.google.com/spreadsheets/d/
1a53W********fpSRXk** /edit#gid=0

後の工程で必要になる情報なので、
控えておくとスムーズです。
同時に、**人に教えてはいけない機密情報**でもあります。
取り扱い注意です！

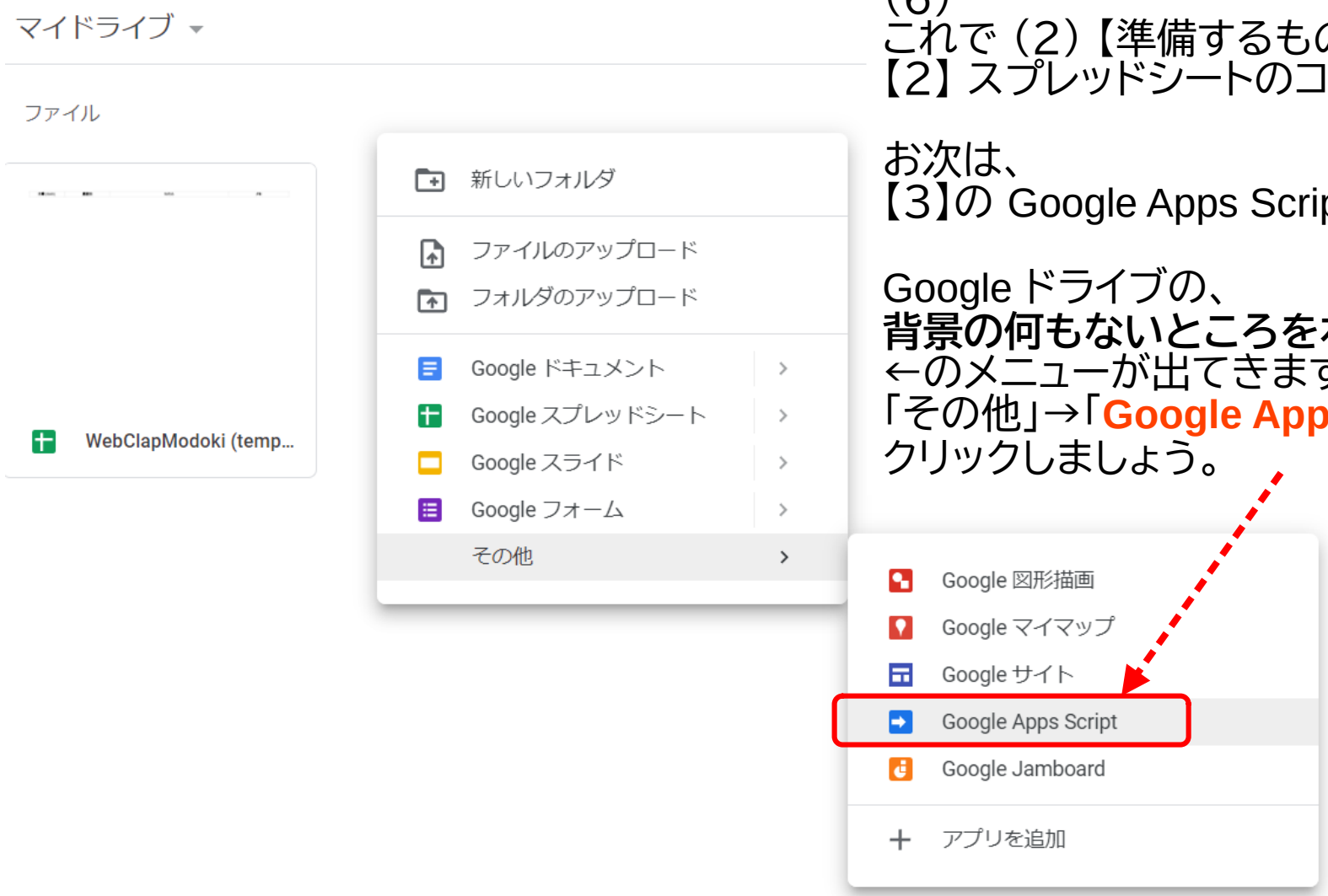
（3.1）Google ドライブ上に必要なファイルを用意（5／23）



(5)
さて、Google ドライブの画面に戻ってみましょう。
ちゃんと「WebClapModeki (template)」のコピーが
表示されていればOKです！

※　本手順書では変えていませんが、先の id でシートを特定しているため、
このファイルの名前は自由に変えて大丈夫です。
「ゴーストの感想」など、ご自身が分かりやすいようにリネームして下さい。

（3.1） Google ドライブ上に必要なファイルを用意（6／23）



（6）
これで（2）【準備するもの】で言及していた
【2】スプレッドシートのコピーが用意できました！

お次は、
【3】の Google Apps Script ファイルを用意します。

Google ドライブの、
背景の何もないところを右クリックすると、
←のメニューが出てきますので、
「その他」→「**Google Apps Script**」を選んで
クリックしましょう。

（3.1）Google ドライブ上に必要なファイルを用意（7／23）

（7）
「Apps Script - 無題のプロジェクト」という画面が、
新しいタブで開いたでしょうか？
この画面が出てきたら、ひとまずOKです。
次のスライドへどうぞ。

（3.1） Google ドライブ上に必要なファイルを用意 （8／23）

（8）
一旦、dullNeko の GitHub リポジトリ上にある WebClapModoki_GAS.txt を開いて下さい。
RAW(テキスト)表示のURL(↓)を開き、Ctrl+A で全選択してから Ctrl+C でコピーして、

https://raw.githubusercontent.com/dullNeko/WebClapModoki-GAS/main/WebClapModoki-GAS.txt

先の Apps Script のエディタに Ctrl+V で **全文を** 貼り付けて、Ctrl+S で上書き保存して下さい。

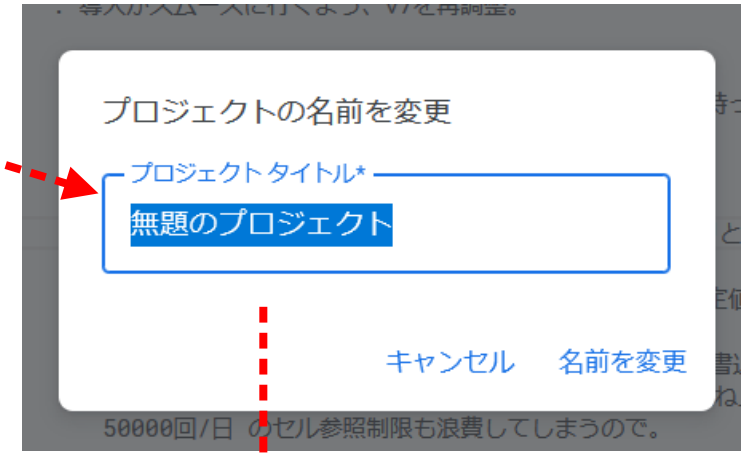
```
//=============================================================================================
// WebClapModoki_GAS_v7
//
// 【作成者】
// dullNeko (https://github.com/dullNeko/WebClapModoki_GAS)
//
// 【更新履歴】
// 2022/12/21 v7  : 公開用に調整した最初のバージョン。
//                  MastodonへのAPI経由でのトゥート機能を試験実装するとともに、
//                  セーフティとしてプロパティサービスを利用しないと動作しないように調整。
// 2022/12/15 v1  : Web拍手代替物がGASで組めるかのテスト版。
//                  とりあえず動作確認できたレベル。
//=============================================================================================


//=============================================================================================
// ★ 参考資料
//=============================================================================================
  /*
    覚えておいた方が良いページなど。
    「直接コードを引用した。」「大いに参考にさせて頂いたページ」等は
    コード周辺に配置して明示するようにしています。
  */

  //=============================================================================
  // ★ GAS全体の制限について
  //=============================================================================
  /*
    ■ Google サービスの割り当て  |  Apps Script  |  Google Developers
    https://developers.google.com/apps-script/guides/services/quotas

    GASは基本的に無償で使用できるサービスですが、
    代わりに、上記URLのページの通り「実行可能量」の制限があります。
    > Apps Script サービスには、1 日あたりの割り当てと、一部の機能に対する制限があります。
    > 割り当てまたは上限を超えると、スクリプトが例外をスローし、実行が停止します。
```

全文コピペ

（3.1） Google ドライブ上に必要なファイルを用意 （9／23）

（9）
タイトルが「無題のプロジェクト」のままでも、動作に支障はない、のですが…
後々、混乱のもとになりやすいので、ここで変えておきましょう。

（10-1）
「無題のプロジェクト」付近にカーソルを持っていくと、
「名前を変更」という表示が出てくるので、
そこで左クリック。



（10-2）
「プロジェクトの名前を変更」
ウィンドウが出てくるので、
「WebClapModoki_GAS_v8」
あたりに変えます。

（10-3）
変え終わったら、
「名前を変更」
をクリックして、
確定させて下さい。

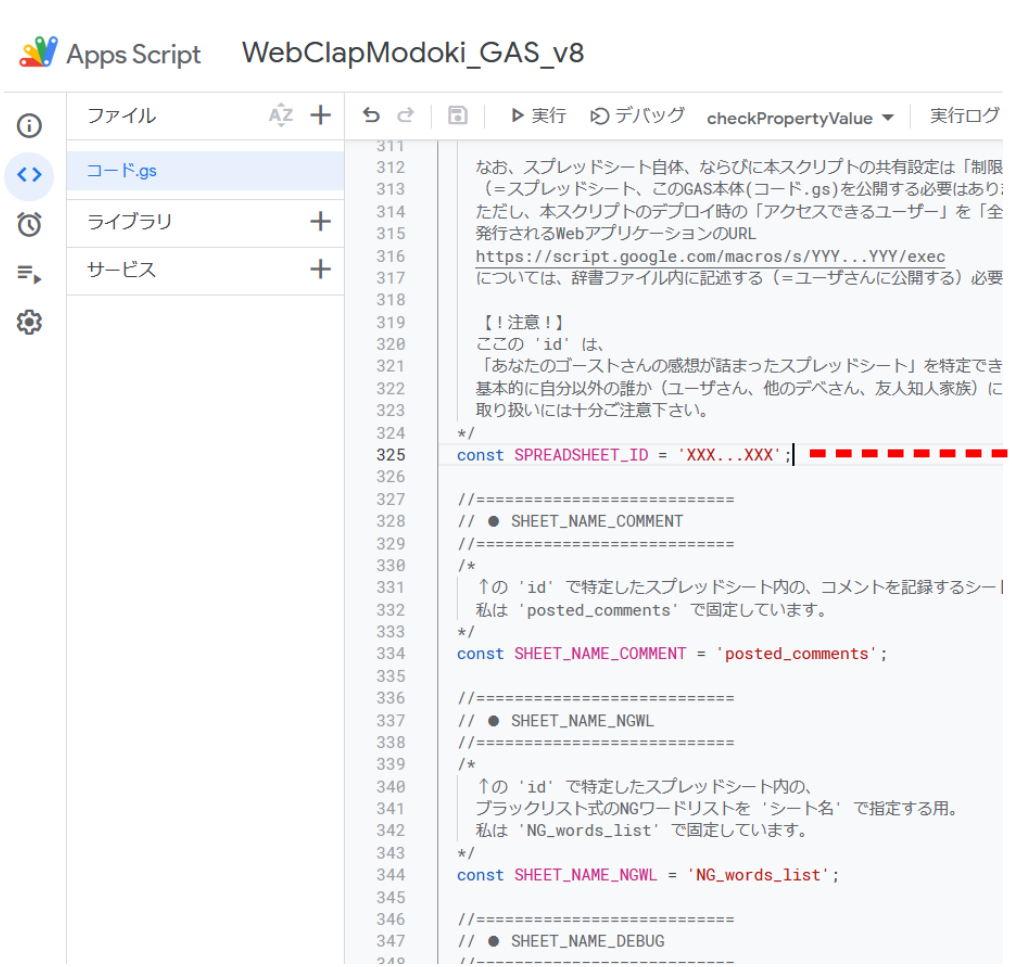
（3.1） Google ドライブ上に必要なファイルを用意 （10／23）

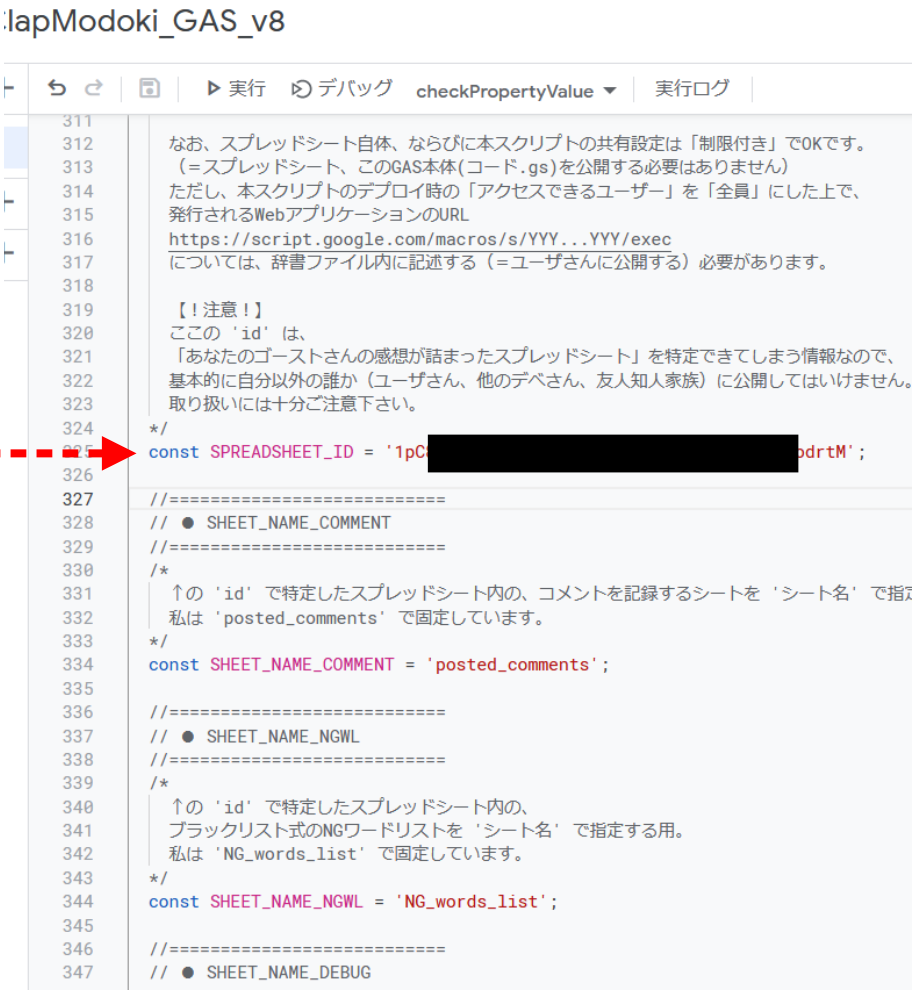
（10）
元の画面に戻って、タイトルが「WebClapModoki_GAS_v8」に変更されていればOKです。

それでは、(4-2)で控えた
「スプレッドシートの id」を、本スクリプトに記入していく、GASの準備段階に移ります。
次のスライドへどうぞ。

（3.2）Google Apps Script の準備／動作確認／設定変更（11／23）

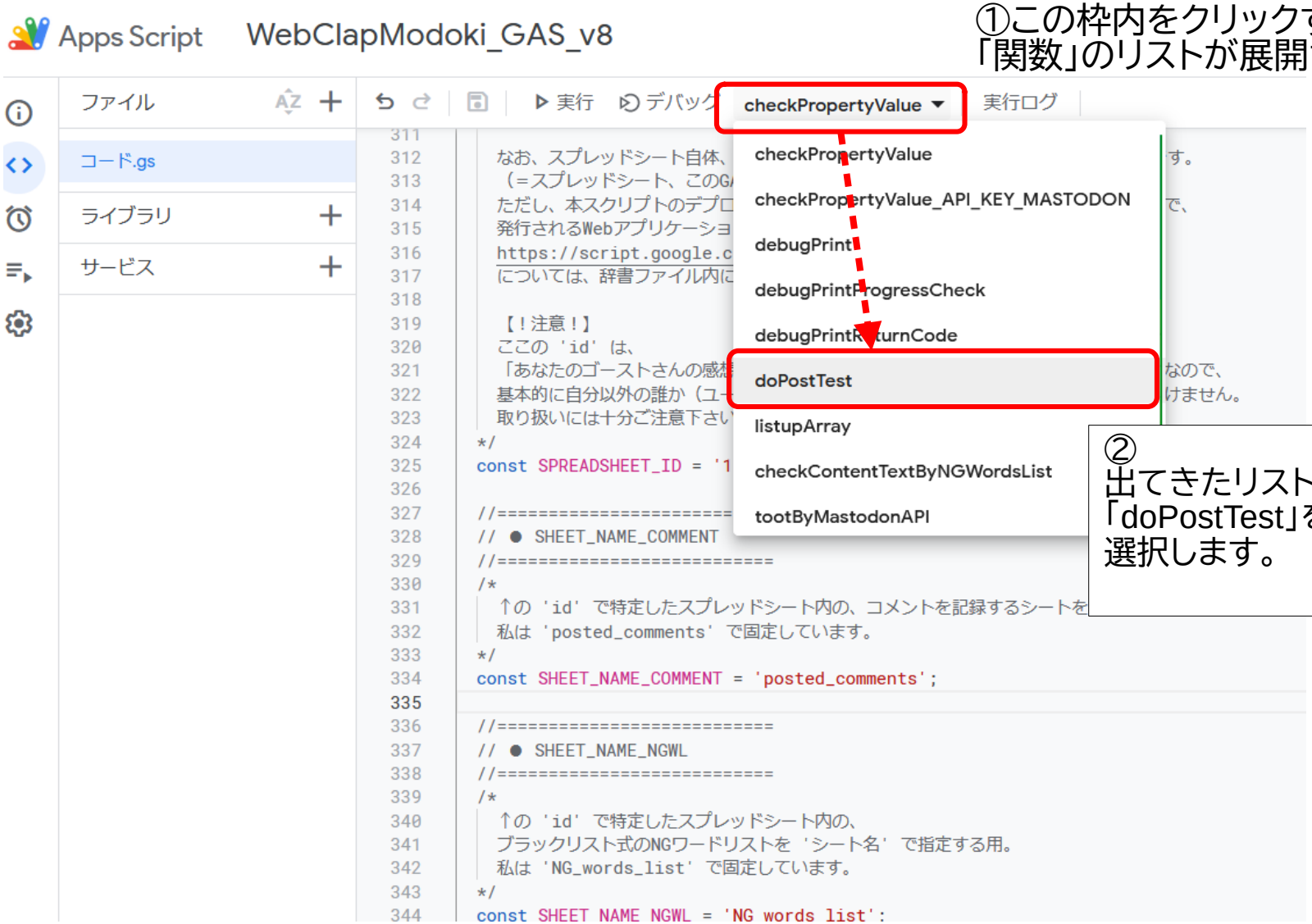
(11)
(4-2)の「スプレッドシートの id」を、本スクリプトの 325行目の 'XXX...XXX' に入れて、
　const SPREADSHEET_ID = '「スプレッドシートの id」';
として書き換えます。
書き換えたら、Ctrl+Sで上書き保存を忘れずに。　　（※先述した通り、機密情報なので一部黒塗りです）





（3.2）Google Apps Script の準備／動作確認／設定変更（12／23）

（12）
これで準備が整いました。
それでは、動作確認をしていきましょう。
「デバッグ」の右にある「関数」のリストから、「doPostTest」を選択します。



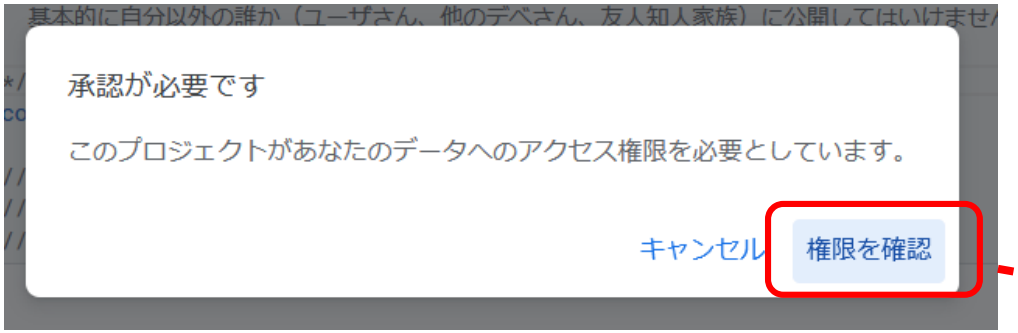
（3.2）Google Apps Script の準備／動作確認／設定変更（13／23）

（13）
選択された関数が「doPostTest」になっていることを確認して、
「実行」ボタンをクリックします。

（3.2）Google Apps Script の準備／動作確認／設定変更（14／23）

（14）
…ここで、記事本文の**【注意喚起】**で記述していた、
「承認が必要です：このプロジェクトがあなたのデータへのアクセス権限を必要としています」
ウィンドウが出てきます。

以降、「アクセス権を与えて、実行を許可する」方法を示します。



（14-1）
「権限を確認」ボタンをクリックします。

（14-2）
（この例では私のアカウントですが）
ご自身のアカウントが表示されるので、
それをクリックして選択します。

（3.2）Google Apps Script の準備／動作確認／設定変更（15／23）

（15）
個人が作成したスクリプトなので、
下記の警告（Googleのチェックを受けていないが良いか？）が出ます…が、
許可しないと実行できないので、先に進みます。

（15-1）
「権限を確認」ボタンをクリックします。

（15-2）
「無題のプロジェクト(安全ではないページ)に移動」
ボタンをクリックします。

（3.2）Google Apps Script の準備／動作確認／設定変更（16／23）

（16）
最終確認画面です。
「許可」をクリックすれば、以下のアクセス権限が与えられて、
スクリプトを実行可能な状態になります。

（16-1）
「許可」ボタンをクリックします。

（3.2）Google Apps Script の準備／動作確認／設定変更（17／23）

（17）
（16）で「許可」をクリックすると、元の画面に自動で戻り、doPostTest関数の実行が行われます。
ここまでで設定ミス等がなければ、
「実行開始」の表示後、数秒内に、「お知らせ 実行完了」表示が出ます。
これで、正常に動作することが確認できたので、次にデプロイを行います。

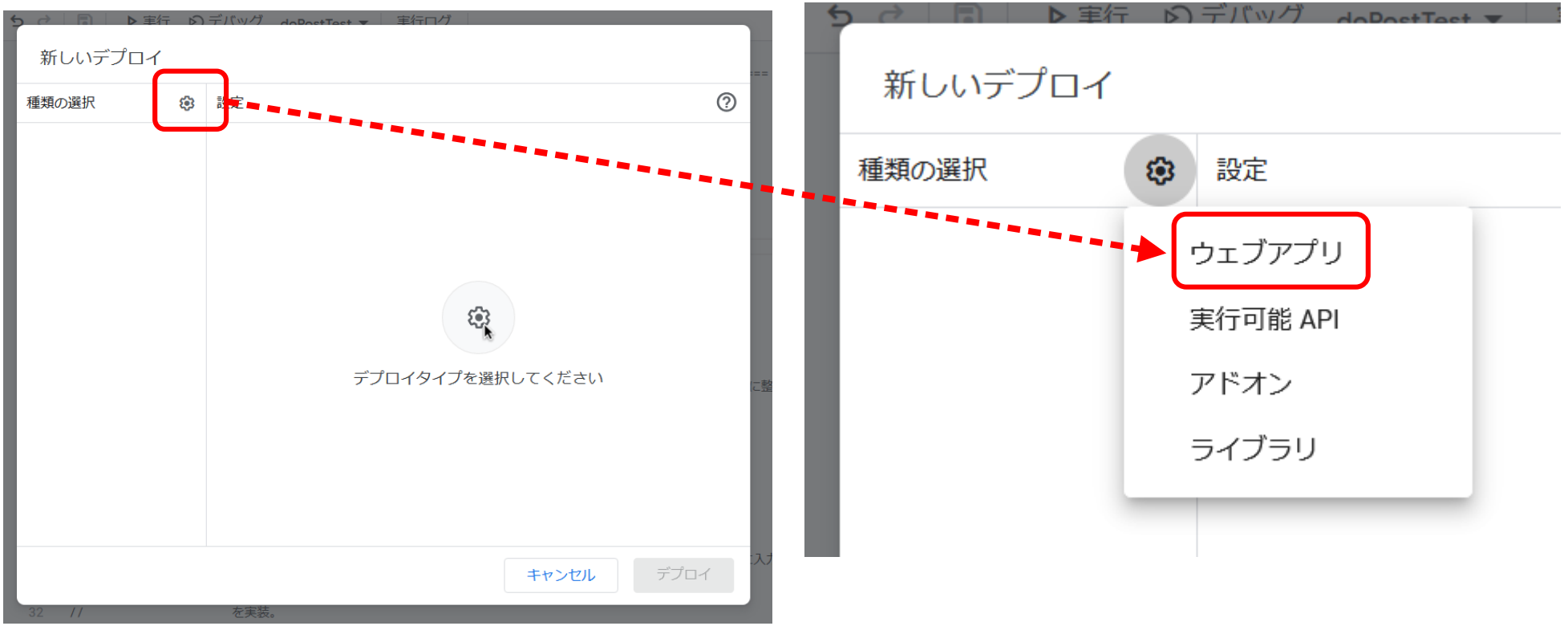
（17-1）
「お知らせ　実行完了」が表示されていればOKです。

（3.3）Google Apps Script のデプロイ／WepアプリURL取得（18／23）

(18)
「デプロイ」ボタンをクリックして、
その中の「新しいデプロイ」をクリックします。

（3.3）Google Apps Script のデプロイ／WepアプリURL取得（19／23）

（19）
「種類の選択」の右にある「歯車」ボタンをクリックし、
その中の「ウェブアプリ」をクリックします。



（3.3）Google Apps Script のデプロイ／WepアプリURL取得（20／23）

（20）
「アクセスできるユーザー」をクリックし、
その中の「全員」をクリックして設定します。

（3.3）Google Apps Script のデプロイ／WepアプリURL取得（21／23）

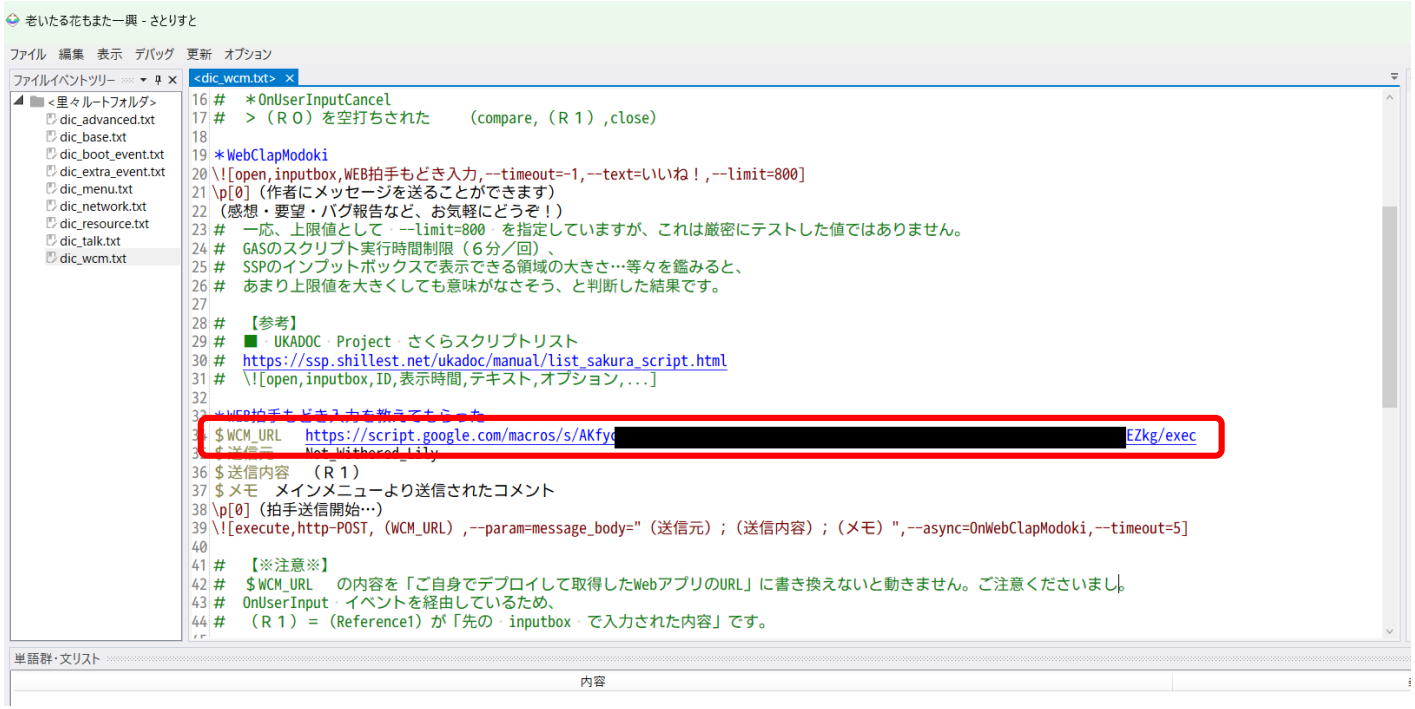
（21）
「デプロイ」をクリックして実行し、
Webアプリとして使用する用のURLを発行します。
「デプロイを更新しました。」が表示されたら、
「ウェブアプリ」のURLを「コピー」し、「完了」ボタンをクリックして閉じて下さい。

（3.4）SSP側でのHTTPリクエストの実装／動作テスト（22／23）

（22）
次の \![execute,http-POST,(WCM_URL),...] 文の(WCM_URL)部分に、
(21)でコピーしたWebアプリ用URLを代入し、
ゴーストさんの辞書内に配置して、呼び出してください。

\![execute,http-POST,(WCM_URL),--param=message_body="(送信元);(送信内容);(メモ)",--async=OnWebClapModoki,--timeout=5]

里々であれば、本プロジェクト内の dic_wcm.txt をフォルダに入れ、
https://github.com/dullNeko/WebClapModoki-GAS/blob/main/dic_wcm.txt
＄WCM_URL　　https://script.google.com/macros/s/YYY~YYY/exec
のURLを、(21)でコピーしたWebアプリ用URLに置き換えて上書き保存し、
＊WebClapModoki を呼び出せばOKです。(送信元)、(メモ)は、お好みで変更してください。



（3.4）SSP側でのHTTPリクエストの実装／動作テスト（23／23）

（23）
実際にゴーストさんから呼び出してみて、
記事冒頭のGIF画像のように、
記録先スプレッドシートに記録されることが確認できたら、
初期設定完了です！お疲れ様でした。