



# 天気予報LINEBot勉強会

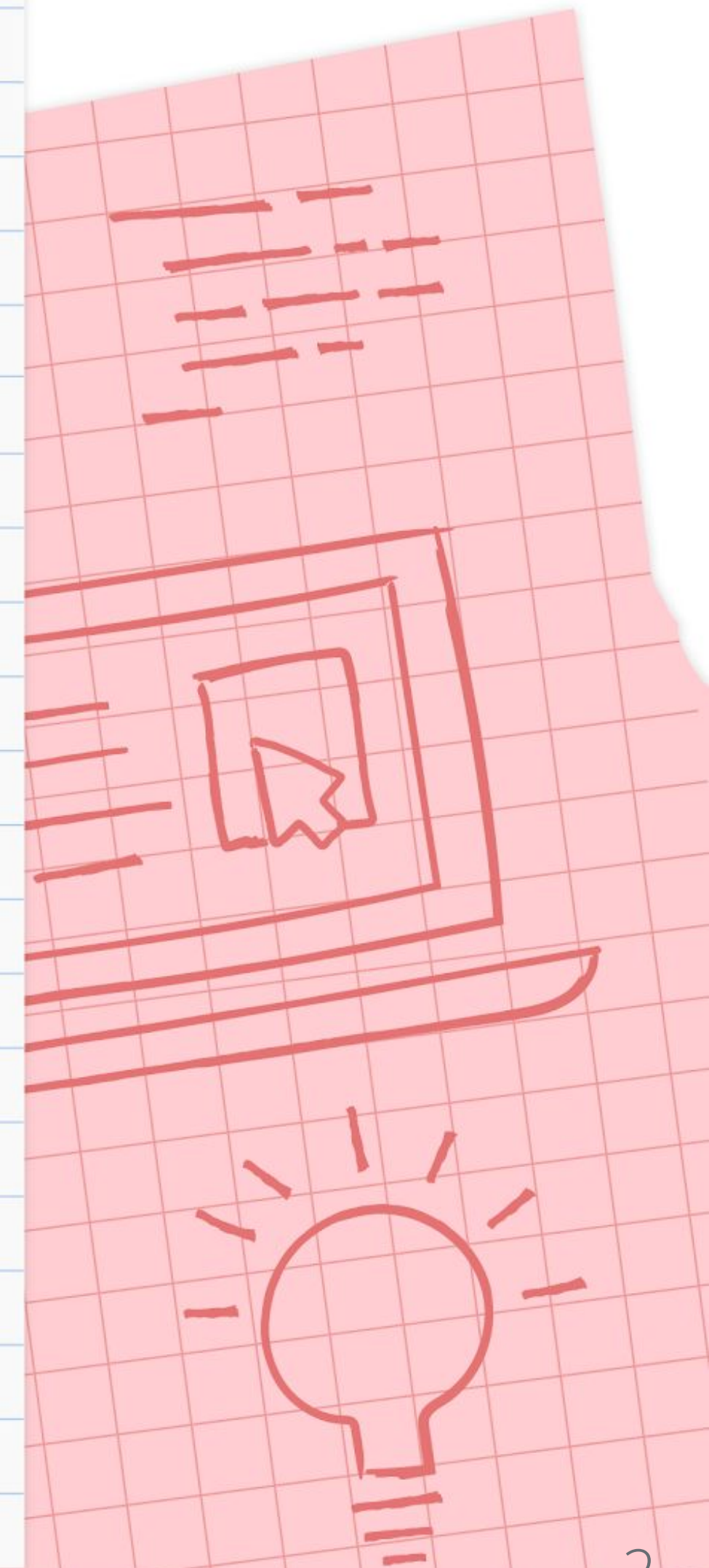
LINE Notify×GASでサクッと完成

Presented by Aoi Hashimoto

```
filterByOrg = filterByOrg ? study.lead_organization === filterByOrg : true
filterByStatus = filterByStatus ? study.status === filterByStatus : true
return (matchStatus) {
  function filterStudies({ studies, filterByOrg, filterByStatus }) {
    return studies.filter(study => filterByOrg && filterByStatus)
```

# 目次

- ❑ 概要
- ❑ GAS
- ❑ 天気API
- ❑ LINE Notify
- ❑ 質問・感想





# 天気予報LINEBot

## 概要

毎朝天気と最高気温を教えてくれるLINEBot。

## <使用ツール>

- GAS
- LINENotify
- 天気予報 API (livedoor 天気互換)

[今日の天気]  
おはようございます!!  
今日は2023-12-06です。

天気:晴のち曇

最高気温:null

19:30

今日



LINE Notify

[今日の天気]  
おはようございます!!  
今日は2023-12-07です。

天気:晴れ

最高気温:20

1:07



Aa



# 天気予報LINEbot

## 実装イメージ



LINE Notify

③送信



GAS

①天気教えて～



②はい！どうぞ



天気予報  
API



# GAS(Google Apps Script)



GAS(Google Apps Script)は、Googleが提供する各種サービスやその他のサービスの自動化／連携を行うためのローコード開発ツール。  
GASでプログラムコードを記述際に使用するスクリプト言語は、JavaScriptをベースに作られている。

# 天気予報 API (livedoor 天気互換)



天気予報 API(livedoor 天気互換)は、気象庁が配信している全国各地の今日・明日・明後日の天気予報・天気詳細・予想気温・降水確率と都道府県の天気概況情報を JSON データで提供している。




# LINE Notify



LINE Notifyは、LINEとWebサービスを連携し、ユーザーが通知を受け取れるサービス。Webサービスと連携すると、LINEが提供する公式アカウント"LINE Notify"から通知が届く。複数のサービスと連携でき、グループでも通知を受信することが可能。



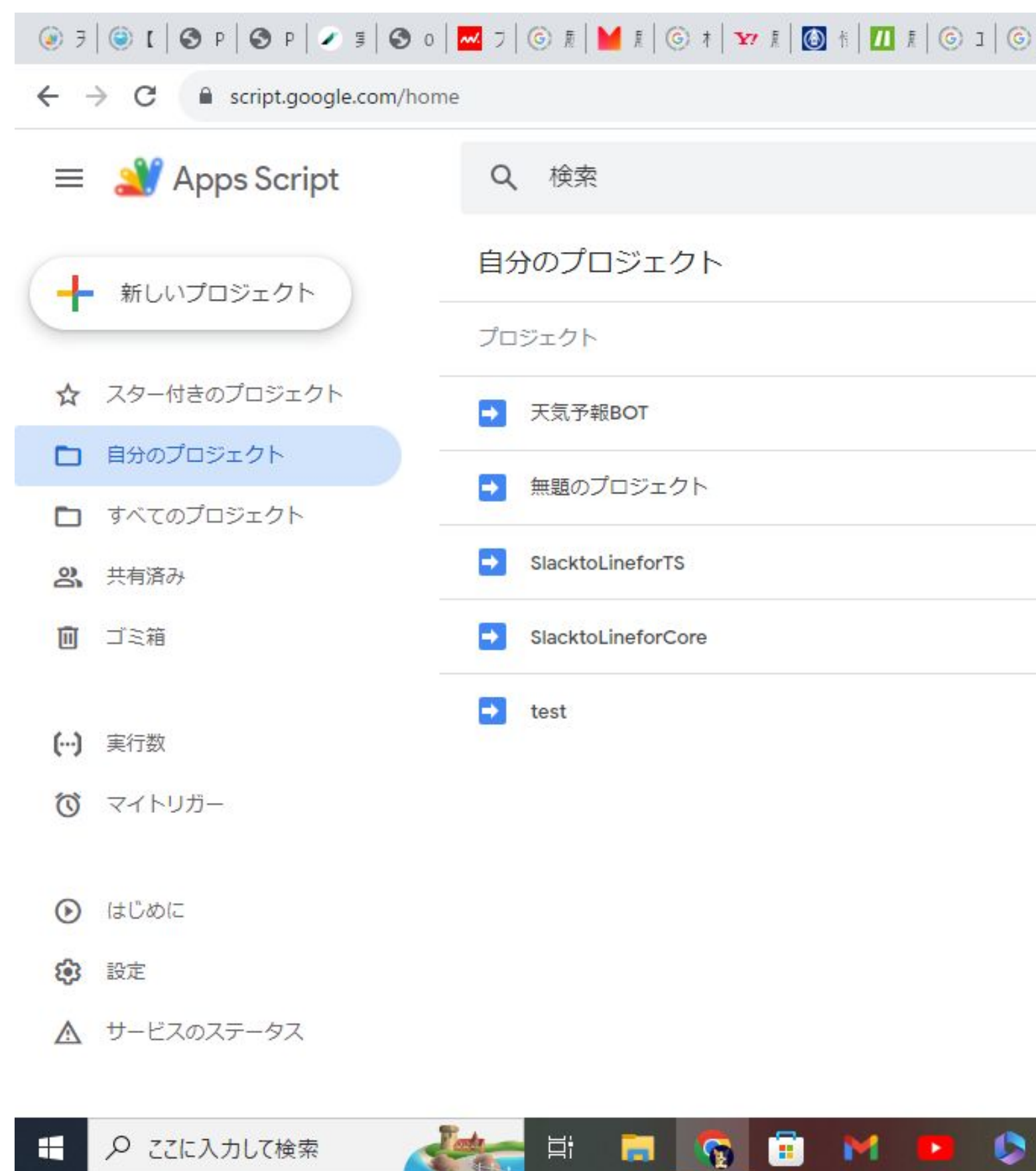


“

8

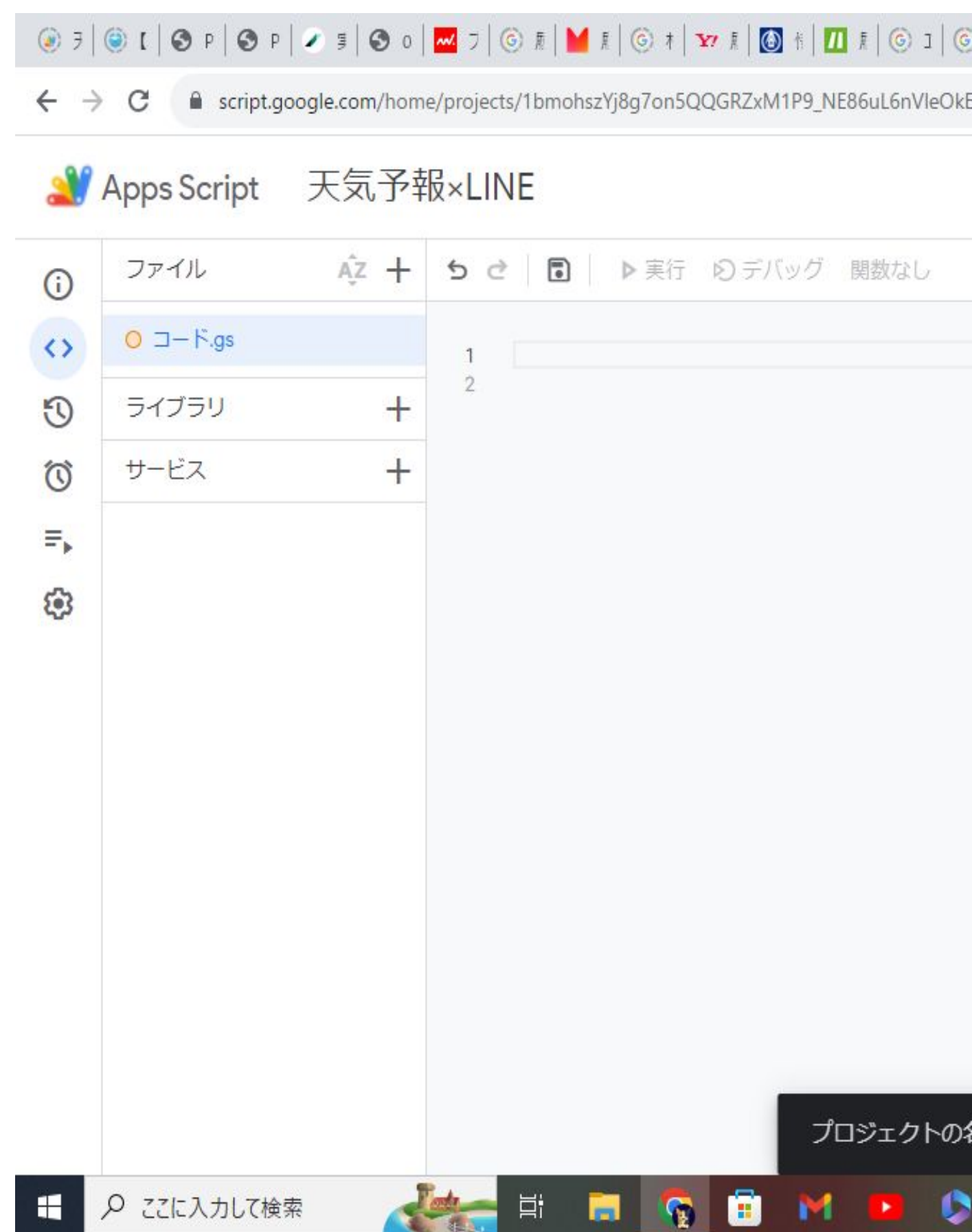


# GASの準備(1/3)



1. <https://script.google.com/home>にアクセス
2. 新しいプロジェクトを選択

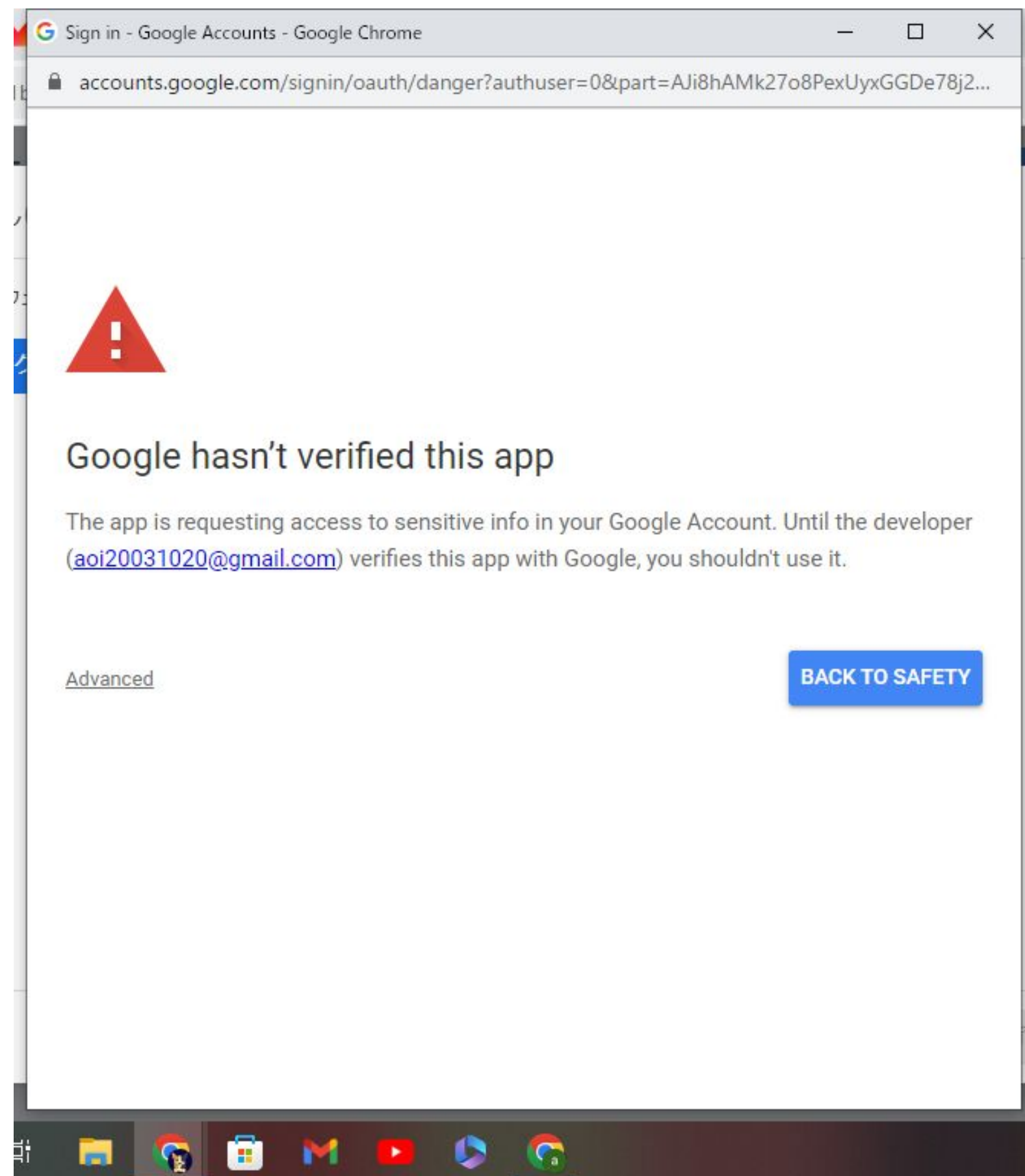
# GASの準備(2/3)



1. 名前をつける
2. 書かれているコードを全て消す



# GASの準備(3/3)



1. デプロイ→新たなデプロイ
2. 種類の選択の歯車→Webアプリを選択
3. 右下のデプロイをクリック
4. アカウントの認証を行う  
※この画面になった場合左下のAdvancedをクリックして進む(エラーでなくセキュリティメッセージ)

# 天気APIの呼び出し(1/3)

```
function GetWeatherFunction(){  
    const API_Data = JSON.parse(Ur1FetchApp.fetch(//ここにURL).getContentText());  
    console.log(API_Data);  
}
```

APIの

URL : "<https://weather.tsukumijima.net/api/forecast/city/130010>"

APIを呼び出して結果を出力




# 天気APIの呼び出し(2/3)

```
let WeatherData = [];  
{  
  WeatherData[0] = API_Data["forecasts"][0]["date"]  
  WeatherData[1] = API_Data["forecasts"][0]["telop"]  
  WeatherData[2] = API_Data["forecasts"][0]["temperature"]["max"]["celsius"]  
}  
console.log(WeatherData);
```

# 天気APIの呼び出し(3/3)

```
message = "\nおはようございます!!\n今日は" + WeatherData[0] + "です。 \n\n天気:" + WeatherData[1] +  
"\n\n" + "最高気温:" + WeatherData[3];  
console.log(message);  
//send(message); //後で使う
```



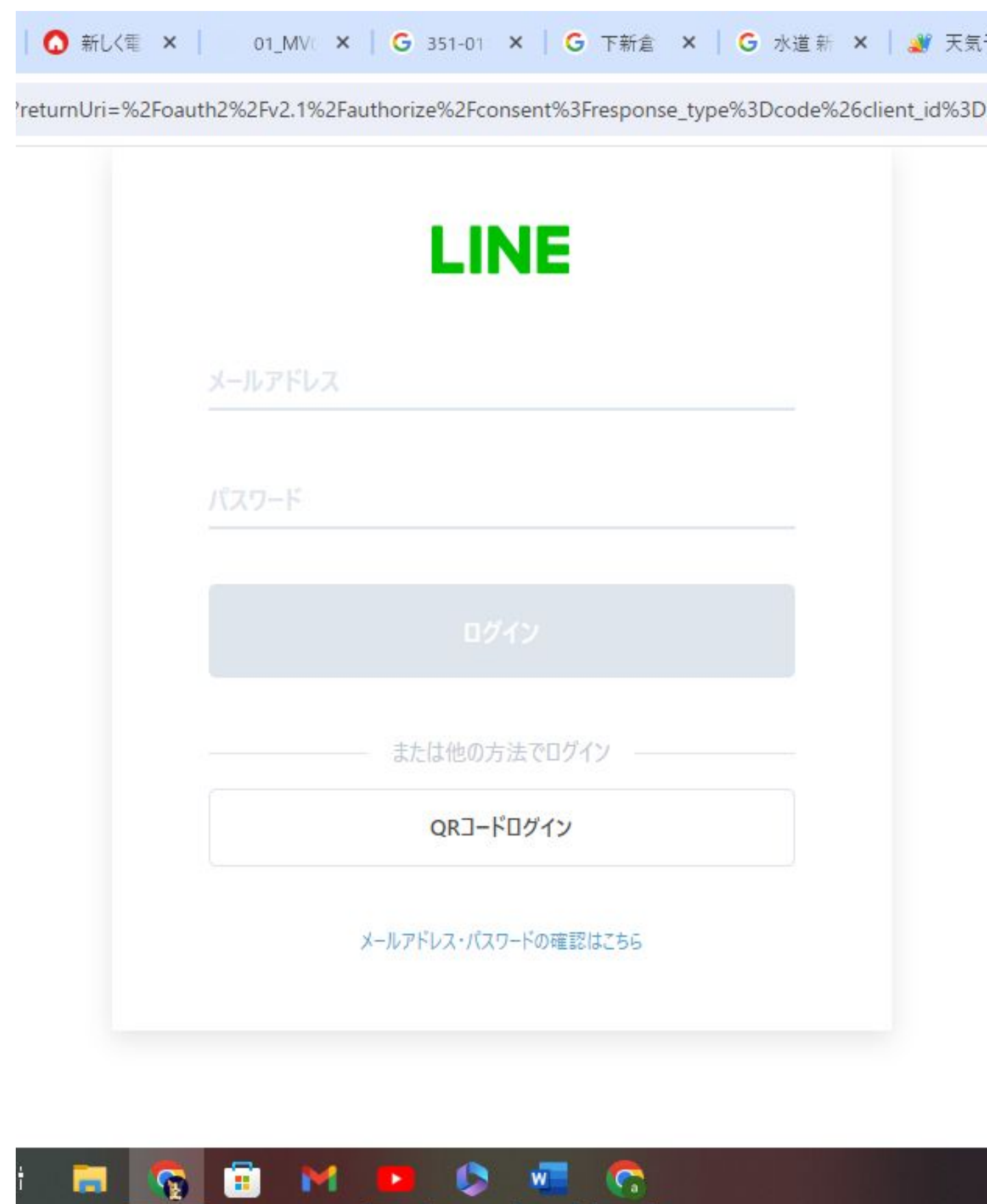


送信するメッセージの完成！！

“

```
function filterStudies({ studies, filterByOrg = false, filterByYear = false }) {
  return studies.filter(study => {
    if (filterByOrg) {
      return study.organizational_unit !== 'Other'
    }
    if (filterByYear) {
      return study.start_year !== 0
    }
    return true
  })
}
```

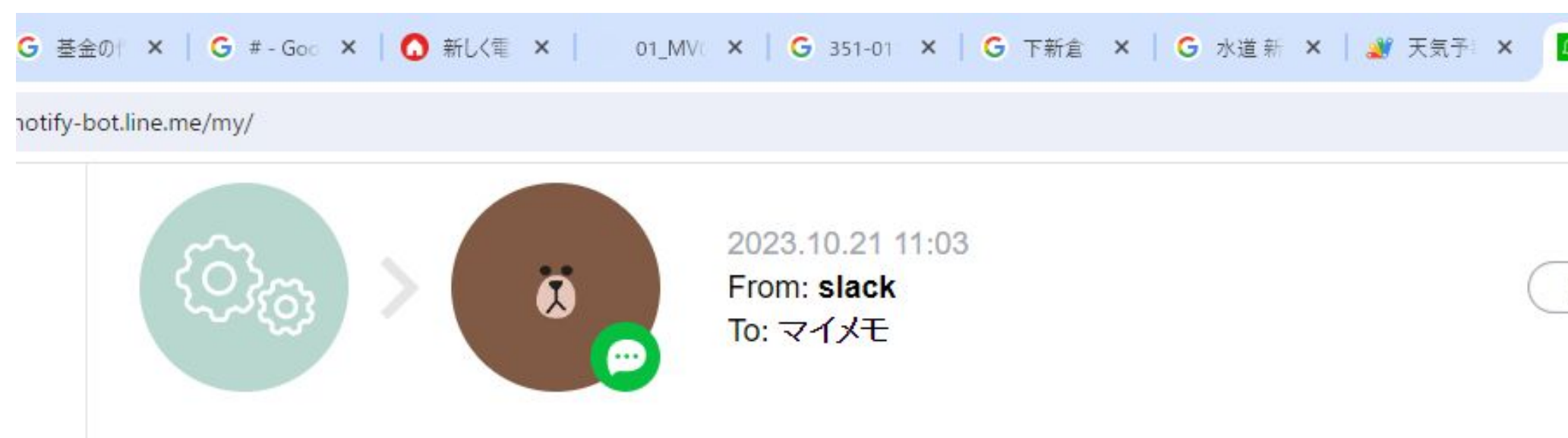
# LINE Notifyの準備(1/3)



1. <https://notify-bot.line.me/ja/>にアクセス
2. 右上からログイン



# LINE Notifyの準備(2/3)



1. 右上アカウント名→マイページに移動
2. ページ下部から「トークンを発行する」

## アクセストークンの発行(開発者向け)

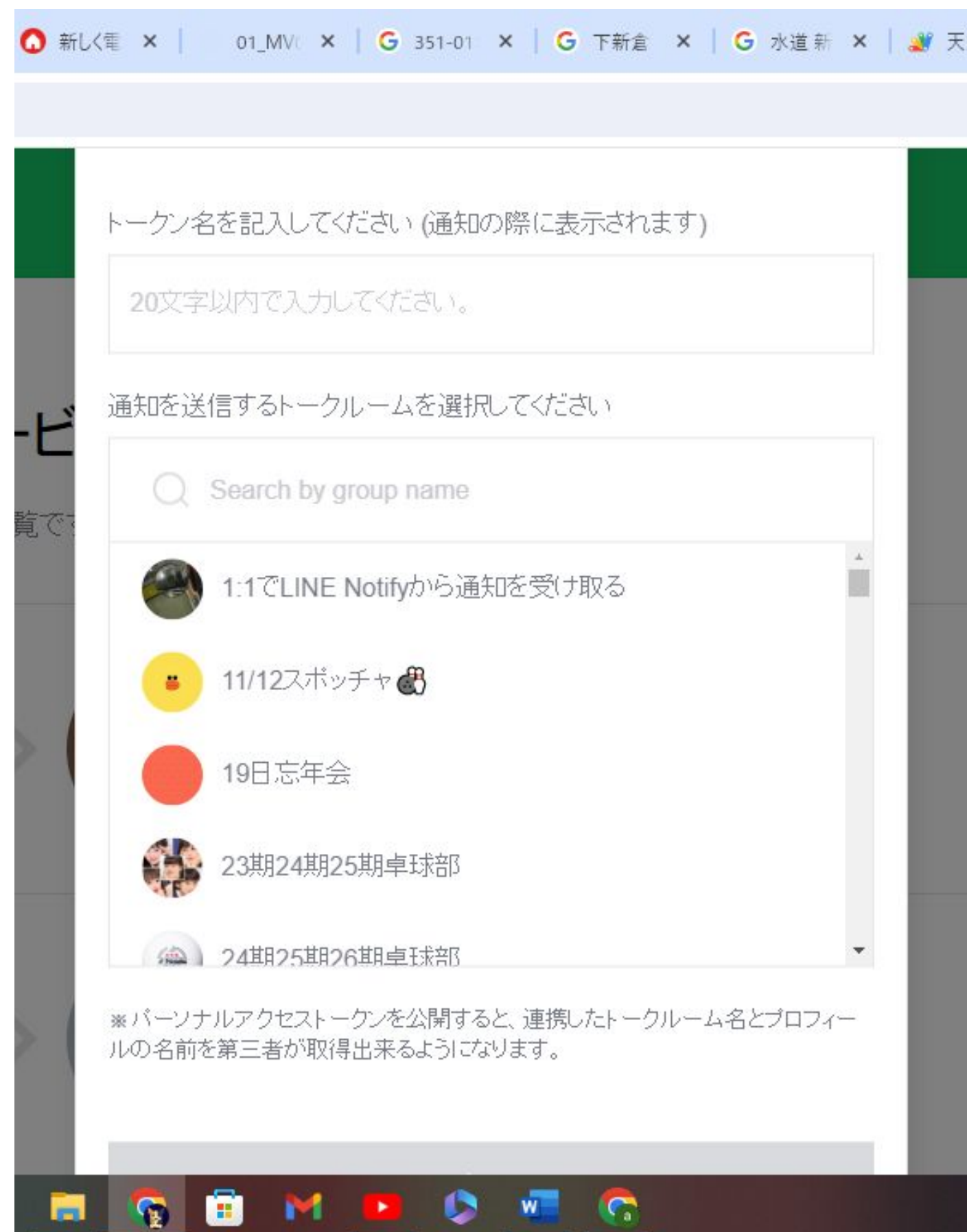
パーソナルアクセストークンを利用することで、Webサービスの登録をせずに通知を設定することができます。

トークンを発行する

LINE Notify API Document



# LINE Notifyの準備(3/3)



1. トークン名をつける
2. 「1:1でLINE Notifyから通知を受け取る」を選択
3. 発行→トークンキーをコピー

※ここで発行したトークンキーは必ず控えておく



# メッセージを送信(1/3)

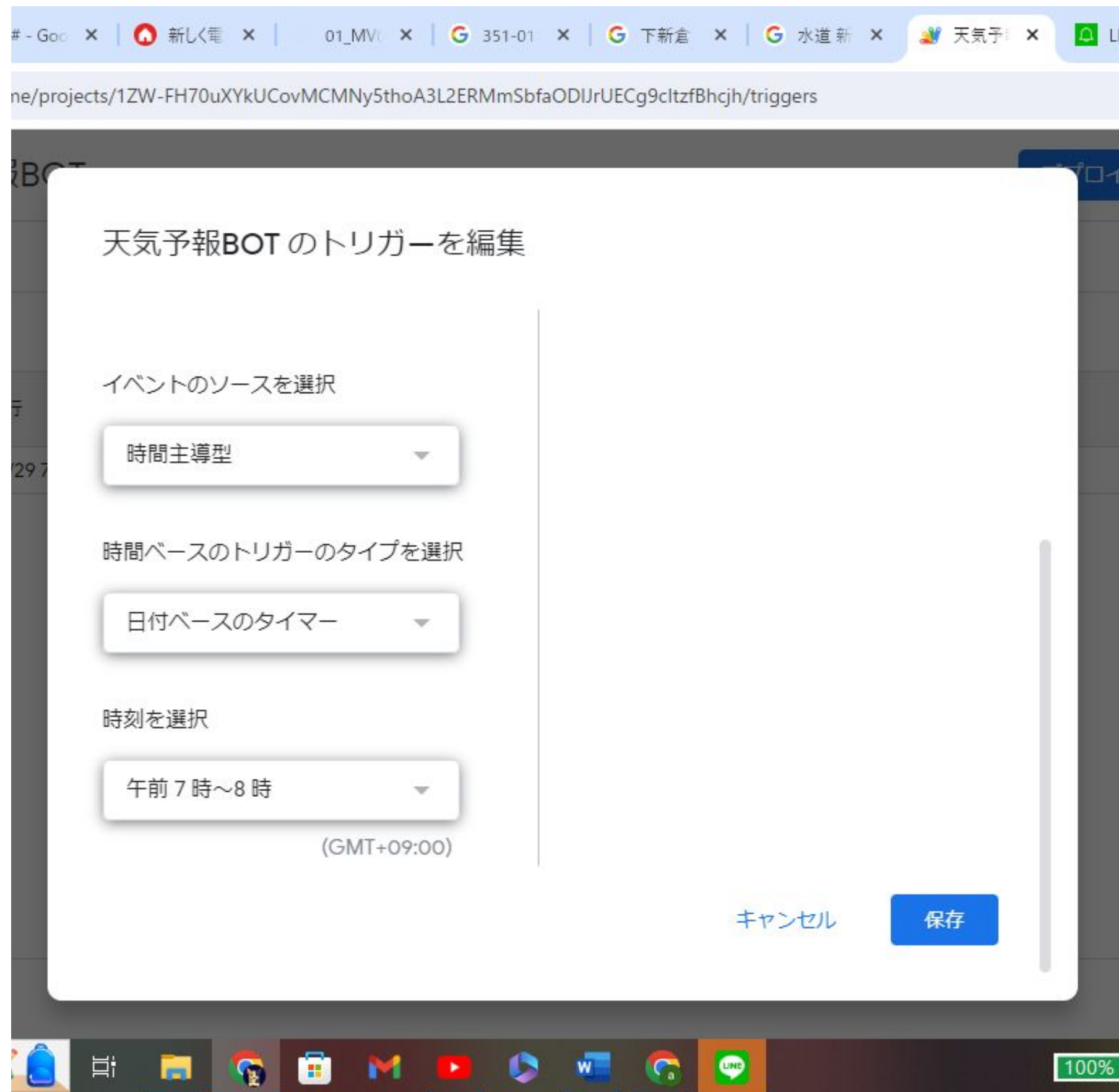
```
function send(message){  
  var token = "トークンキー";  
  var op =  
    {  
      "method" : "post",  
      "payload": "message=" + message,  
      "headers":{"Authorization" : "Bearer " + token}  
    };  
  UrlFetchApp.fetch("https://notify-api.line.me/api/notify",op);  
}
```

# メッセージを送信(2/3)


1. プロジェクトを保存 (Ctrl + s)
2. 実行をクリック
3. メッセージがLINEに送信されれば成功



# メッセージを送信(3/3)



1. 左側トリガーを選択
2. トリガーを追加
3. 時刻を選択→好きな時間を選択
4. 保存



“



# 質問・感想

お疲れ様でした！！  
よい春休みを～