地域課題×ICT で深化する生物育成の新たなアプローチ

～自動計測と WEB アプリで実現するデータの蓄積と可視化～

飯綱町立飯綱中学校　髙野健人

Ⅰ　はじめに

令和の日本型教育は、急速に変化する社会や新たな技術環境に対応しつつ、生徒一人一人の可能性を最大限に引き出すことを目指している。その中で、技術・家庭科教育は、生徒が実践的かつ体験的な学び、探究する力を育む重要な役割を担っている。しかし、従来の指導方法では、生徒が主体的に課題を見いだし、その解決に向けて深く考察する機会や、学びの過程を客観的に評価し共有するプラットフォームが十分に整っていないという課題が存在する。そのため、生徒が実践的なスキルや持続可能な社会づくりに必要な資質・能力を効果的に身に付けることが難しい現状がある。

本研究の目的は、技術・家庭科教育において追究のテーマ設定及び ICT を活用した WEB アプリを導入することで、令和の日本型教育が目指す能力の育成を効果的に支援することである。 Google フォームや micro:bit を活用したシステムを開発・導入し、日々の学習活動をデジタル化、データを共有・分析する環境を整えることで、深い学びを促進することを目指す。

２　研究仮説

本研究の仮説は、技術分野の生物育成の授業場面における深い学びを促進するための追究のテーマ設定と ICT の活用に焦点を当てたものであり、その具体を下記に示す。

(1) 問題発見・探究活動の促進

地域の農業観光課から得た実際のデータを基にした身近な問題を取り上げることで、生徒は「なぜそのような状況にあるのか」、「どのように改善できるのか」といった問いを自然にもち、探究活動を主体的に展開することが期待される。

(2) 創造的な課題解決能力の育成

micro:bit を用いたセンシング技術やデータ管理をしながら、生徒は ICT の活用の基本原理と応用方法を実践的に学ぶことができる。それにより、根拠のある解決策の提案が可能となり、持続可能な社会づくりに向けた創造的な思考力が育まれると考えられる。

(3) 協働的な学びの充実

WEB アプリを介してクラス全体で同じ情報を共有することで、話し合いにおいて相互の主張に対する根拠を理解しやすくなる。そのようなデータに基づいた根拠のある意見を交わし合うことで他者の視点を取り入れながら探究していきやすくなる。

(4) 計画性・予測能力の強化

天気予報機能や環境データの活用により、生徒は事前に状況の変化を想定し、計画的な管理や対応する能力が向上する。

以上の仮説が正しければ、本研究で導入したテーマ設定及び ICT の活用は、生物育成の授業場面において、生徒が身近な問題を糸口に探究活動を主体的に進め、創造的な思考力を発揮して持続可能な社会へとつ

ながる問題解決に取り組むための有効な実践となると考えられる。

３　研究内容（実践の紹介）

本研究では、生物育成の授業場面において、身近な問題を糸口にしたテーマ設定と ICT を活用した WEB アプリの導入を通じ、生徒が探究活動を主体的に進め、創造的な思考力を育むための具体的な手立てを検証した。以下では、その実践内容とシステムの特徴を示す。

(1) 身近な問題を糸口にしたテーマ設定

地域の農業観光課から得た「農業従事者の減少」、「(農業従事者の) 高齢者・若者の割合」に関する情報（図１）を学習の糸口として提示した。これにより、生徒は「なぜ農業従事者が減少しているのか」「持続可能な農業はどのような条件で成立するのか」といった身近でありながら社会的意義のある問題を自然に認識できるようになった。このテーマ設定によって生徒は単なる知識習得にとどまらず、現実の課題に根差した探究活動へと踏み込む環境が整った。

農業従事者の変化（飯綱町）

年	農業経営体数
平成 12 年（2000 年）	1,335
令和２年（2020 年）	819

高齢者・若者の割合（飯綱町）

年	農業経営体数	29 歳以下	60 歳以上
平成 12 年（2000 年）	1,335	0.07%	68.84%
令和２年（2020 年）	819	0.12%	81.93%

飯綱町農業観光課より

図１　飯綱町に農業に関する情報

(2) システムの特徴

① Google フォームによる記録の一元管理

生徒は毎日の観察記録を Google フォーム（図２）を通じて入力する。フォームには「観察日」、「記録者」、「管理内容」、「作物の写真」の項目が含まれており、直感的にデータを入力できるよう設計されている。

② センサによる自動計測

micro:bit を通じて温度、照度、水分のデータを自動計測する（図３）。これにより、データの一貫性と正確性が確保される。

③ Google スプレッドシートへの自動保存

Google フォームからの入力および micro:bit からの計測データは、Google スプレッドシートに自動保存され、データの信頼性を高める。

図２　生徒が入力する Google フォーム

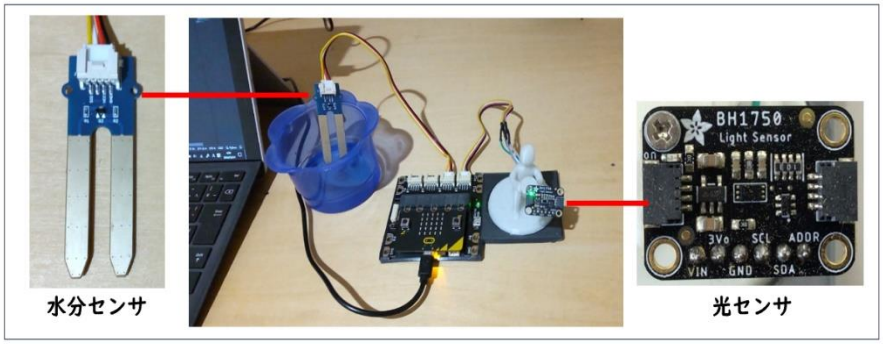


図３　各種センサ（温度センサはマイクロビット内蔵）

WEB アプリ等のマニュアルは
こちらからダウンロード可能

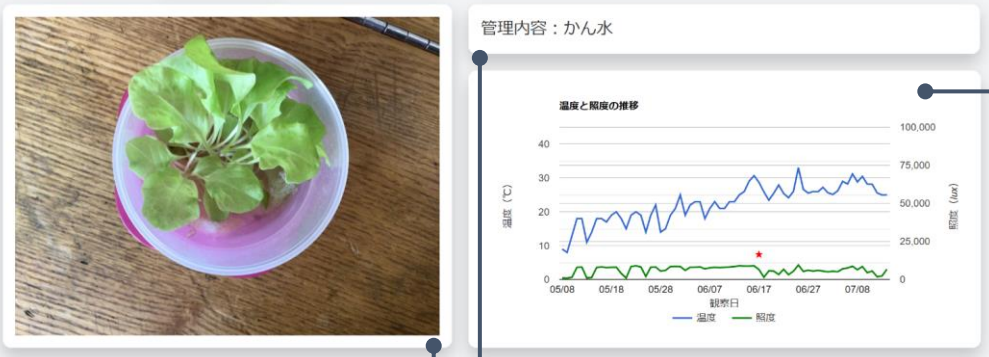
④　WEB アプリ

PC への遠隔アクセス：
　選択した栽培方法に応じて、対応する PC の GoogleMeet を返して遠隔でその場の状況を知ることができる。

リアルタイムグラフの表示：
　選択した栽培方法に応じて、スプレッドシートに蓄積されたデータを基に、温度、照度、水分の情報を随時グラフ化する。

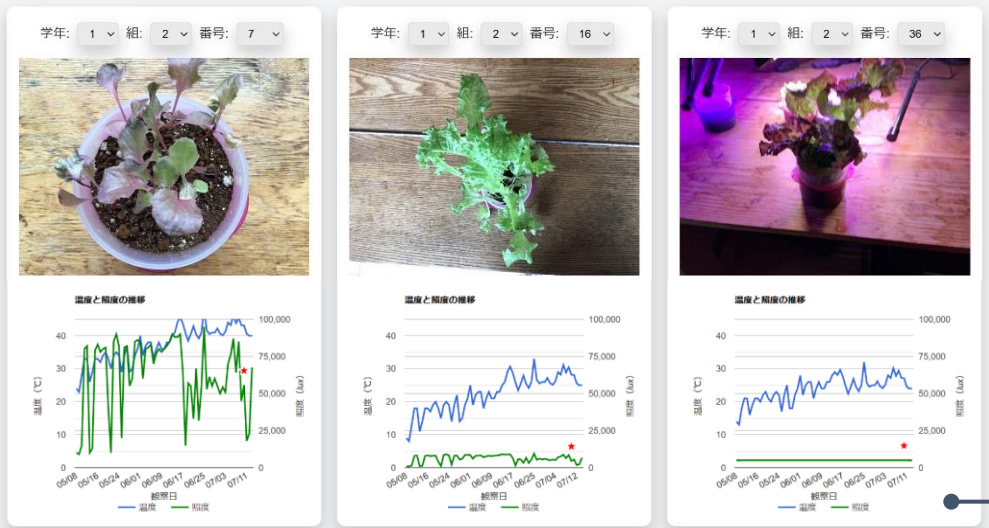
水位メッセージ：
　水位が基準値を下回ると警告メッセージを表示する。水分センサを使用しない場合には非表示にすることも可能。

天気データの取得と表示：
　スプレッドシートから取得した天気データを基に、その日の天気と週間天気予報を表示する。



温度と照度の経過：
　自分の栽培方法の温度と照度の累積データをグラフ化する。
★は指定した日付を表している

観察記録の表示：
　日付を指定すると、その日の作物の写真と管理内容を表示する。



観察記録の比較表示：
　日付を指定すると、その日の作物の写真と温度と照度の経過を表示する。最大 3 つの栽培方法を一覧比較することができる。

(3) テーマ設定及び ICT の活用による効果

① 問題発見・探究活動の促進

ア 現状調査・情報収集から実践的な栽培体験へ

生徒は最初に、土耕栽培、太陽光型水耕栽培、人工光型水耕栽培といった多様な栽培方法を調べ、実際に栽培体験を行った。その中で、「どうすれば農業の生産性を向上させたり、農業従事者を増やせたりするのか？」といった問いを持ち続けながら追究を行う様子が見られた。

イ 探究への動機付け

身近な問題を出発点としたことにより、生徒は自分の学びが地域や社会に直結していることを実感し、探究活動に主体的に取り組む姿が見られた。ある生徒は、「農業従事者の減少や若者離れが深刻だと聞いていたが、データを見て本当に大変な問題なんだと分かった。自分たちで何ができるか考えたくなった」と述べ、テーマ設定が動機付けとなっている様子が見られた。

② システム導入による効果

ア 創造的な課題解決能力の育成

WEB アプリを介して収集・共有された温度、照度、水分などのデータを分析する中で、生徒は科学的根拠に基づいた判断や改善策を立てられるようになった。生徒たちから「水位が下がっている警告が出ているよ。休み時間に見に行った方がよさそうじゃない？」、「温度が高すぎるからハウスの扉を開けに行こう」といった具体的な対話や「グラフを見て、もっと光が必要そうなときは人工光型水耕栽培が有利だと気付いた。」といった対比的な発言を引き出した。

イ 協働的な学びの充実

WEB アプリ上でクラス全体が作物の写真や温度、照度などの環境データを共有できることで、生徒同士が課題に対する気付きや考えを簡単に交換できるようになった。あるグループが「土耕栽培は管理が多いが、時間と知識があれば地域の高齢者でも続けやすい方法がある」と意見を出し、別のグループが「スマート農業の導入により手間を軽減する」と反応した。このように共有されたデータを基に生徒たちのアイデアを発展させ、協力して問題解決に向けた具体的な対策検討をする姿が見られた。「友達が出したアイデアを自分の考えと組み合わせてもっとよくできるかと思った」という生徒の声からも、協働学習を通じた発想の相互作用が確認された。

ウ 計画性・予測能力の強化

天気予報機能と過去の観察データを参照することで、生徒は事前に状況の変化を想定し、計画的な管理や計画を立てるようになった。「土曜日は雨が降るから、今日は屋根のある所に移動させよう」と話す生徒や、「日照時間が長くなる予報だから、朝から太陽光のよく当たる場所に移動させよう」といった具体的な計画を立てる生徒たちの会話があった。

4 おわりに

本研究では、身近な問題を糸口としたテーマ設定と、ICT を活用した WEB アプリの導入により生物育成の授業場面における深い学びをする支援する検証を行った。その結果、生徒たちはただの知識習得に留まらず、現実の社会課題に基づいた問題発見や探求活動を主体的に行い、創造的な思考力を発揮し、持続可能な社会につながる課題解決へと取り組む姿があった。特に、データの自動計測、環境データの共有、天気予報機能の組み合わせが、生徒の協働学習や計画行動を促進し、創造的な思考力や計画性、予測能力を高める効果があることが確認された。これらの成果は、令和の日本型教育が深い学びを実現する上で、身近な問題を糸口にしたテーマ設定と ICT の活用が有効であることを示唆している。

Glowlog セットアップマニュアル

目次

1. 必要なものの準備
2. Python のインストール
3. Visual Studio Code のインストール
4. 必要な Python ライブラリのインストール
5. Google Sheets API の設定
6. Google スプレッドシートの準備
7. Python スクリプトの準備と編集
8. Python スクリプトの実行
9. トラブルシューティング

---

1. 必要なものの準備

事前に以下のものを準備してください：

- **コンピューター**：Windows 10 または Windows 11 を推奨。
- **インターネット接続**：ソフトウェアのダウンロードや設定に必要です。
- **Google アカウント**：Gmail や Google ドライブを利用するためのアカウント。
- **シリアルデバイス**：micro:bit、データを送信するデバイス。
- **USB ケーブル**：シリアルデバイスとコンピューターを接続するため。
- **Python 3.x**：プログラミング言語 Python の最新版。
- **Visual Studio Code (VS Code)**：プログラムを編集するためのテキストエディタ。

---

2. Python のインストール

2.1. Python のダウンロード

1. **ウェブブラウザを開く**：Google Chrome や Microsoft Edge などを使用します。
2. **Python 公式サイトにアクセス**：
   - アドレスバーに以下の URL を入力して Enter キーを押します：
   - https://www.python.org/downloads/
3. **Python の最新版をダウンロード**：
   - ページ中央にある黄色いボタン「**Download Python 3.x.x**」をクリックします（例：「Download Python 3.13.1」）。

2.2. Python のインストール

1. **インストーラーの実行**：
   - ダウンロードフォルダに移動し、ダウンロードしたファイル（例：「**python-3.13.1-amd64.exe**」）をダブルクリックして実行します。
2. **インストール設定**：
   - **重要！** インストーラーの下部にある「**Add Python 3.x to PATH**」に**チェックマークを入れます。

- 「Install Now」ボタンをクリックします。

3. **インストールの完了**：
   - インストールが完了するまで待ちます。
   - 「Setup was successful」と表示されたら「Close」ボタンをクリックします。

2.3. Python の動作確認

1. **コマンドプロンプトの起動**：
   - スタートメニューの検索バーに「cmd」と入力し、「**コマンドプロンプト**」をクリックして起動します。
2. **Python のバージョン確認**：
   - コマンドプロンプトに以下のコマンドを入力して Enter キーを押します：
   - 正しいバージョンが表示されれば、Python のインストールは成功です。

---

3. Visual Studio Code のインストール

Visual Studio Code (VS Code)は、プログラミング初心者からプロフェッショナルまで幅広く利用されている無料のコードエディタです。以下の手順でインストールします。

3.1. VS Code のダウンロード

1. **ウェブブラウザを開く**。
2. **Visual Studio Code 公式サイトにアクセス**：
   - アドレスバーに以下の URL を入力して Enter キーを押します：
   - https://code.visualstudio.com/
3. **ダウンロードボタンをクリック**：
   - 「Download for Windows」ボタンをクリックして、インストーラー（例：「VSCodeUserSetup-x64-1.XX.X.exe」）をダウンロードします。

3.2. VS Code のインストール

1. **インストーラーの実行**：
   - ダウンロードフォルダに移動し、ダウンロードしたファイル（例：「VSCodeUserSetup-x64-1.XX.X.exe」）をダブルクリックして実行します。
2. **セットアップウィザードに従う**：
   - **「Next」**をクリックして進みます。
   - **利用規約に同意**し、「Next」をクリックします。
3. **インストールオプションの選択**：
   - **「Add to PATH」にチェックを入れる**（推奨）。これにより、コマンドラインから code コマンドを使用できます。
   - その他のオプション（例えば、右クリックメニューに追加するオプションなど）は必要に応じて選択し、「Next」をクリックします。
4. **インストール開始**：
   - 「Install」ボタンをクリックしてインストールを開始します。
   - インストールが完了するまで待ちます。

5. **インストールの完了**：
    - 「**Launch Visual Studio Code**」にチェックを入れ、「**Finish**」ボタンをクリックします。

3.3. VS Code の初期設定

1. **初回起動**：
    - VS Code が起動すると、初回設定のウィンドウが表示される場合があります。「**Select a theme**」などの設定は後から変更可能です。
2. **Python 拡張機能のインストール**：
    - **拡張機能タブ**を開く：
        - 左側のサイドバーにある四つのブロックが集まったアイコン（**Extensions**）をクリックします。
        - または、キーボードショートカット「**Ctrl + Shift + X**」を押します。
    - **Python 拡張機能を検索**：
        - 検索バーに「**Python**」と入力します。
    - **Microsoft 製の Python 拡張機能をインストール**：
        - 「**Python**」という名前で提供されている Microsoft 製の拡張機能を見つけ、「**Install**」ボタンをクリックします。

3. **Python インタプリタの選択**：
    - VS Code の左下にある「**Python インタプリタ**」の表示をクリックします。
    - インストールした Python のバージョン（例：「**Python 3.13.1**」）を選択します。

---

4. 必要な Python ライブラリのインストール

Python スクリプトを実行するために、以下のライブラリをインストールします：

- pyserial：シリアル通信を行うためのライブラリ。
- gspread：Google Sheets API と連携するためのライブラリ。
- oauth2client：Google API の認証を行うためのライブラリ。

4.1. コマンドプロンプトでのインストール

1. **コマンドプロンプトを開く**：
    - スタートメニューの検索バーに「**cmd**」と入力し、「**コマンドプロンプト**」をクリックして起動します。
2. **ライブラリのインストール**：
    - 以下のコマンドを順番に入力し、Enter キーを押します：
    - pip install pyserial
    - pip install gspread
    - pip install oauth2client

- 各コマンドの実行後、インストールが完了するまで待ちます。エラーメッセージが表示されなければ成功です。

4.2. インストールのトラブルシューティング

- **pip が認識されない場合**：
  - Python のインストール時に「**Add Python to PATH**」にチェックを入れていない可能性があります。
  - 再度 Python をインストールし、インストーラーで「**Add Python to PATH**」にチェックを入れてください。
- **インターネット接続の確認**：
  - インターネット接続が不安定だとライブラリのインストールが失敗することがあります。接続状況を確認してください。

---

5. Google Sheets API の設定

Python スクリプトから Google スプレッドシートにアクセスするために、Google Sheets API の設定を行います。

5.1. Google Cloud Platform でプロジェクトを作成

1. **ウェブブラウザで Google Cloud Platform にアクセス**：
   - URL を入力：
   - https://console.cloud.google.com/
   - Google アカウントでログインします。
2. **新しいプロジェクトを作成**：
   - 画面上部の「**プロジェクトの選択**」をクリック。
   - 「**新しいプロジェクト**」をクリック。
   - プロジェクト名を入力（例：「**MeasurementProject**」）。
   - 必要に応じて組織やロケーションを選択し、「**作成**」ボタンをクリックします。
3. **プロジェクトを選択**：
   - 作成したプロジェクトが自動的に選択されない場合は、再度「**プロジェクトの選択**」から選択します。

5.2. Google Sheets API の有効化

1. **API とサービスのライブラリにアクセス**：
   - 左側のメニューから「**API とサービス**」→「**ライブラリ**」をクリックします。
2. **Google Sheets API を検索して有効化**：
   - 検索バーに「**Google Sheets API**」と入力。
   - 検索結果から「**Google Sheets API**」をクリック。
   - 「**有効にする**」ボタンをクリックします。

5.3. サービスアカウントの作成

1. **認証情報の作成**：

- 左側のメニューから「**API とサービス**」→「**認証情報**」をクリックします。
- 上部の「**認証情報を作成**」ボタンをクリックし、「**サービス アカウント**」を選択します。

2. **サービスアカウントの詳細入力**：
   - **サービスアカウント名**を入力（例：「measurement-service-account」）。
   - ID は自動生成されるのでそのままで OK です。
   - 必要に応じて説明を追加し、「**作成して続行**」をクリックします。
3. **サービスアカウントの役割を設定**：
   - 「**役割の選択**」で「**プロジェクト**」→「**編集者**」を選択します。
   - 「**続行**」をクリックします。
4. **サービスアカウントのキーを作成**：
   - 「**キーを作成**」セクションで「**キーの追加**」をクリックします。
   - 「**キータイプ**」は「**JSON**」を選択。
   - 「**作成**」をクリックします。
   - **JSON キーが自動的にダウンロード**されます。このファイルを安全な場所に保存します
   - （例：C:¥Users¥Administrator¥Desktop¥measurement¥measurement-420905-6981cadefd35.json）。

5.4. サービスアカウントのメールアドレス確認

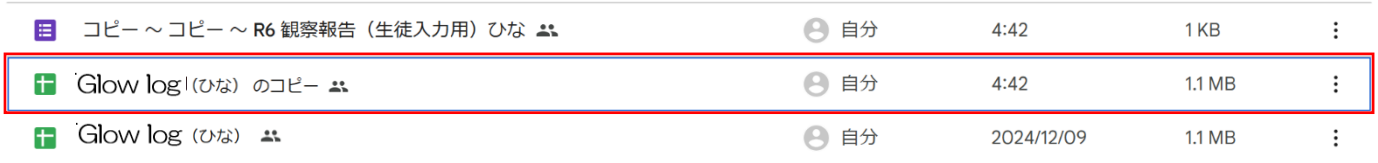
1. **サービスアカウントの一覧を表示**：
   - 左側のメニューから「**IAM と管理**」→「**サービスアカウント**」をクリックします。
2. **メールアドレスの確認**：
   - 作成したサービスアカウントの「**E メール**」欄に表示されているメールアドレスをメモします例：measurement-service-account@measurementproject.iam.gserviceaccount.com

---

6. Google スプレッドシートの準備

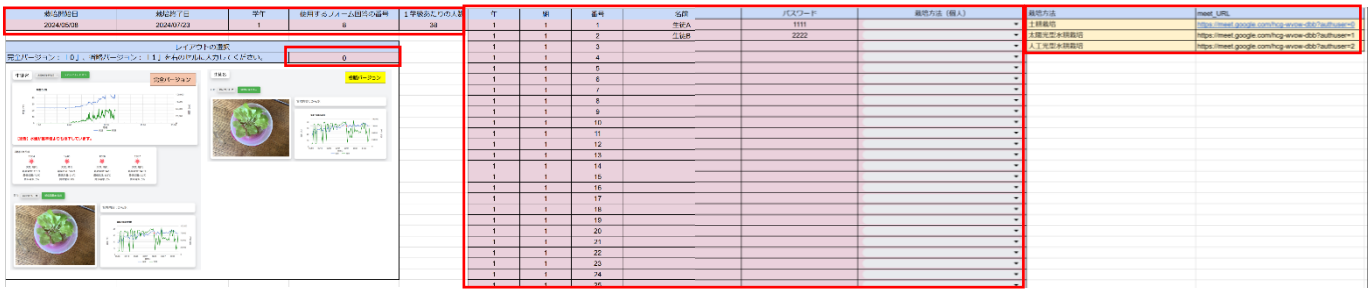
6.1. スプレッドシートの作成

1. **Google ドライブにアクセス**：
   - URL を入力：
   - https://drive.google.com/drive/u/0/folders/1HztKu6TxJJ0LtgH4L7oXd-SEnTdseOpD
   - 「Glow log（ひな）」、「R6 監察報告」を右クリックして「コピーを作成」を選択する。
   - 生成された「コピー」を右クリック→整理→移動によりシートをマイドライブに移してください。その操作なしにマイドライブにコピーされる場合もあります。



2. **スプレッドシートに名前を付ける**：
    - 左上のシート名をクリックし、適当な名前を付けます（例：「**データ収集用シート**」）。

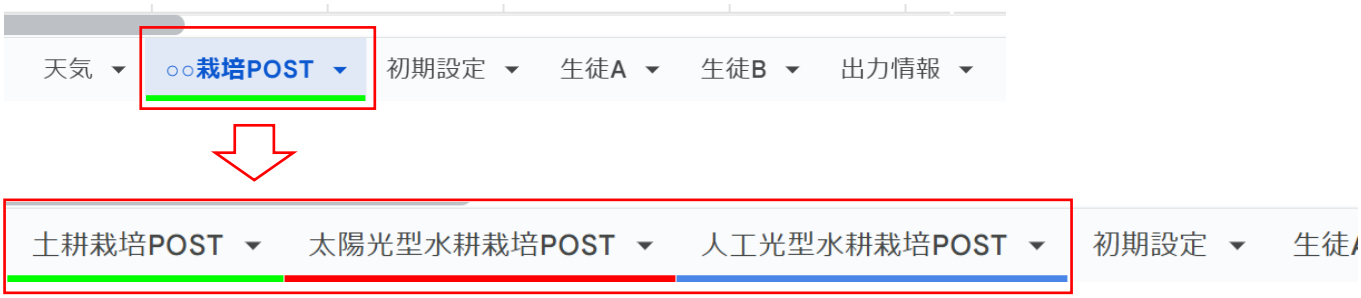
6.2. シート名「初期設定」の作成



1. **シート名「初期設定」の赤枠に必要事項を入力する**：
    - 栽培方法が１つの場合でもＬ列に入力をしてください。
    - 完全バージョンを使用しない場合にはＭ列の「meet_URL」への入力は必要ありません。

6.3. シートの設定

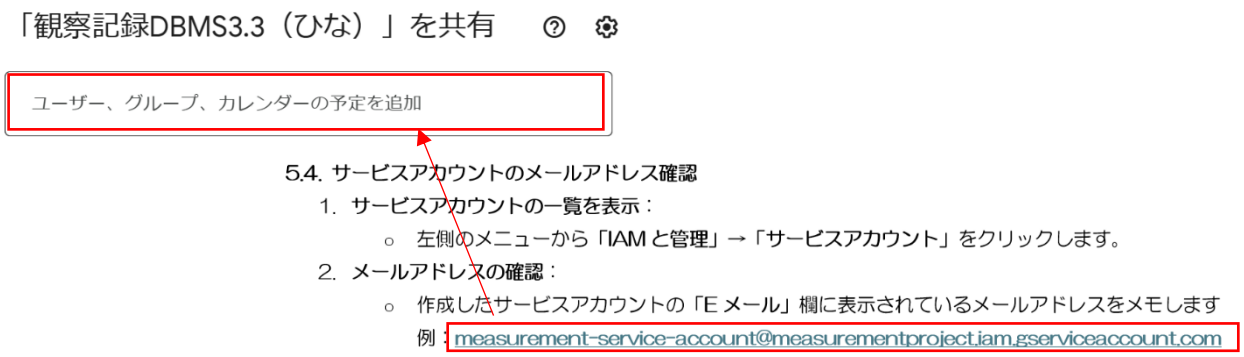
1. 栽培方法 POST を複製する：
    - 「〇〇栽培 POST」を複製し、実際に行う栽培方法の名前の POST を作成する。
      ＊シート名「初期設定」のＬ列と名称を完全一致させること。



6.4. スプレッドシートの共有設定

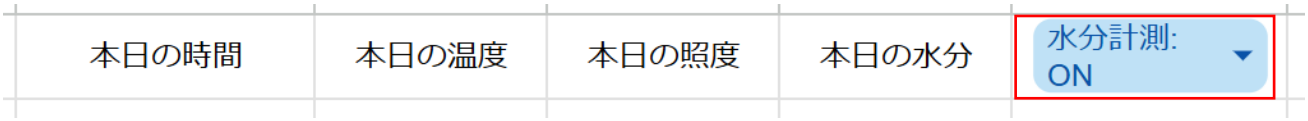
1. 「共有」→「リンクを知っている全員」→「編集者」をクリックする。

2. 「5.4. サービスアカウントのメールアドレス確認」でメモをしたアドレスを貼り付けて共有をする。



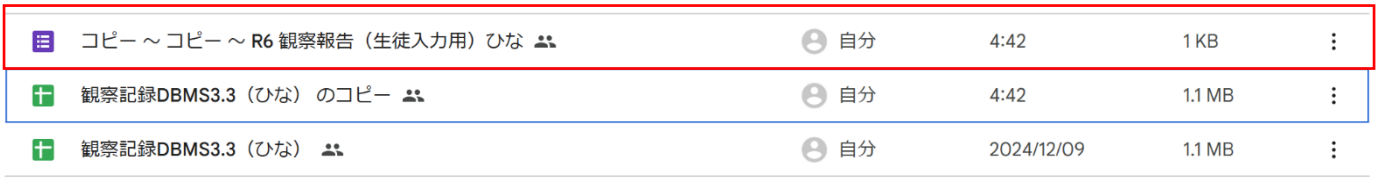
6.5. 水分計測の設定

1. 栽培方法の POST シートを開くと K1 セルに「水分計測 ON」or「水分計測 OFF」を選択できるの
で、必要な栽培方法のみ ON にしてください。

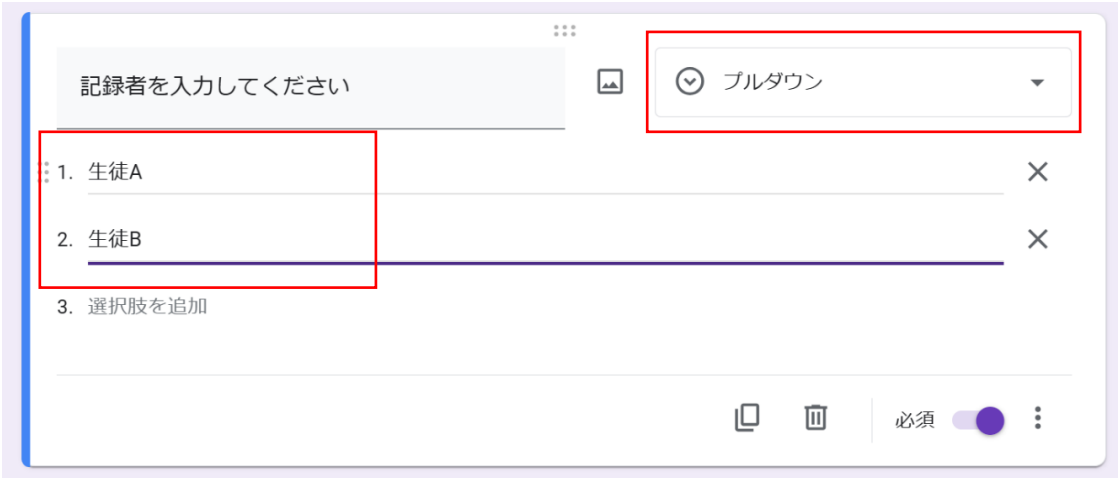


6.6. Google フォームの設定

1. Google フォームのひな型を編集　＊質問項目の順番を変更しないでください
   - 「6.1.1Google ドライブにアクセス」時に生成された Google フォームについてもマイドライブに移動させる。



   - 記録者を「プルダウン」に設定し、シート名「初期設定」の I 列の名前を貼り付ける。

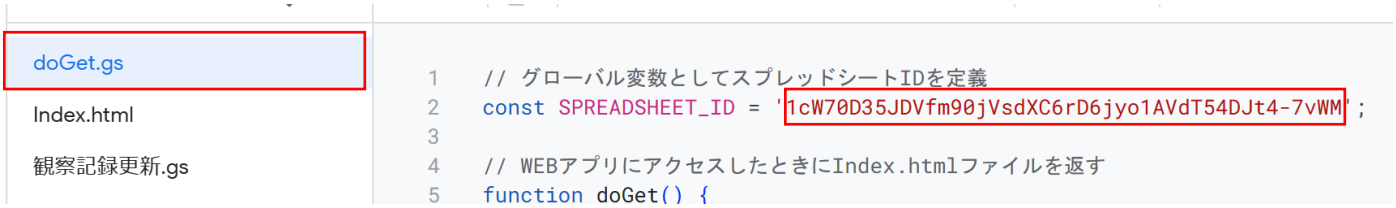


2. 生徒に配付するURLを作成
    - 「送信」→「リンクアイコン」→「コピー」をクリックすると、クリップボードにURLがコピーされる。

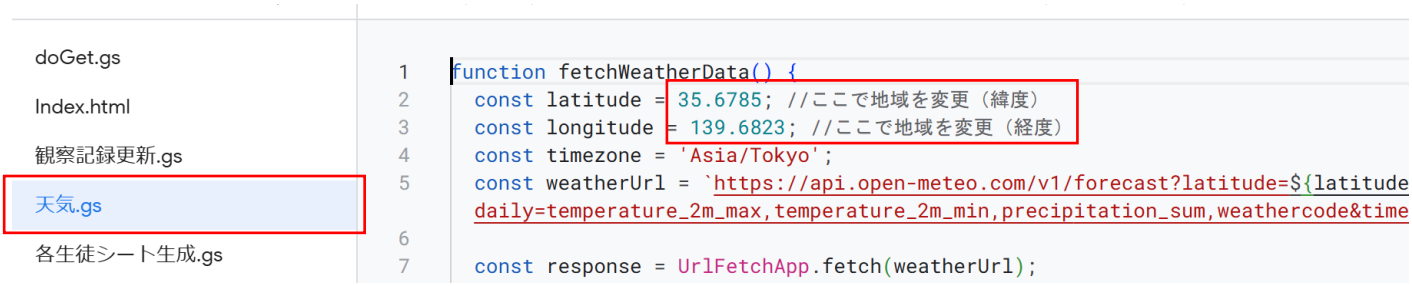
6.6. App Script の設定

1. App Script を開く：
    - 「拡張機能」→「App Script」の順番で開く

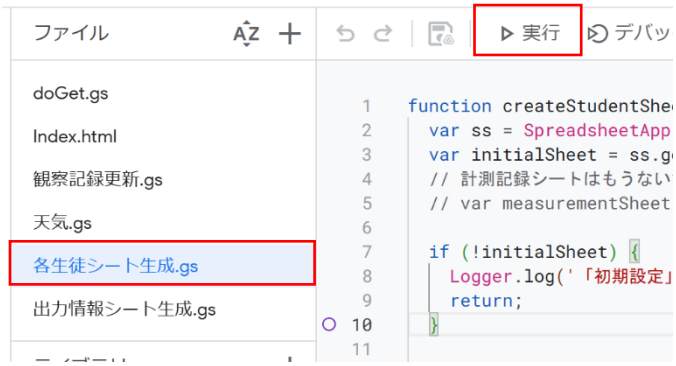
2. スプレッドシートIDの貼り付け：
    - doGet.gsにコピーしたスプレッドシートのIDをコピーして貼り付ける。



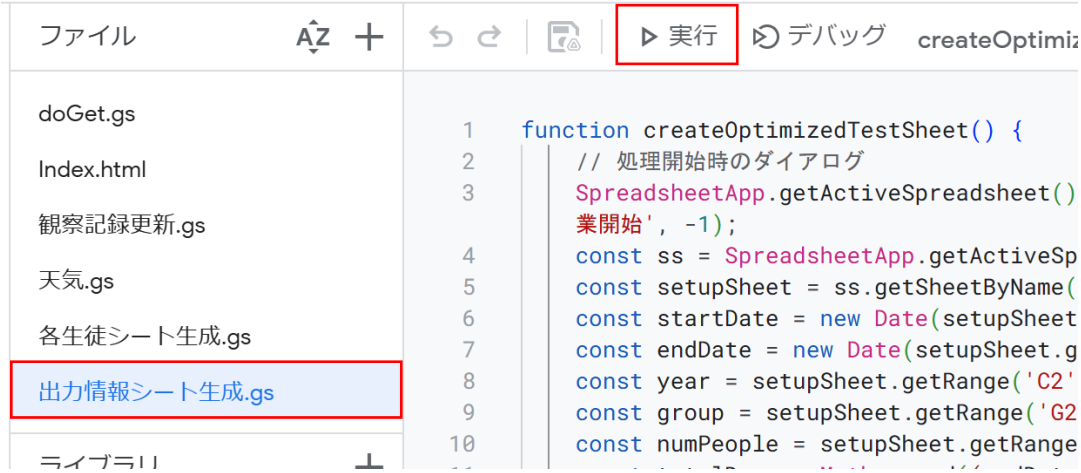
3. 地域情報の追加（完全バージョン選択時のみ）：
    - 天気.gs に授業をする地域の経度と緯度を入力する。



4. 各生徒シートの生成：
    - 各生徒シート生成.gs を選択し、実行をクリックする。



5. 出力情報シートの生成：
    - 出力情報シート生成.gs を選択し、実行をクリックする。



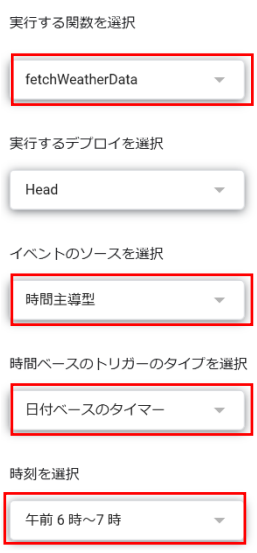
6. トリガーの設定：

- 時計のアイコンをクリック→「トリガーを追加」をクリックする。

- 下図のような設定にして「保存」する。

「summarizeAllData」

「fetchWeatherData」



7. デプロイ：

- 「デプロイ」→「新しいデプロイ」をクリックする。

- アクセスできるユーザーが「全員」であることを確認して「デプロイ」をする。

＊新規の場合、承認をする必要あり。

- 生成された URL を生徒

に配付する。

---

7. Python スクリプトの準備と編集

7.1. スクリプトファイルの作成

1. **Visual Studio Code を開く**：
   - スタートメニューから「**Visual Studio Code**」を検索し、クリックして起動します。
2. **新しいファイルを作成**：
   - メニューバーの「**File**」→「**New File**」をクリックします。
   - または、キーボードショートカット「**Ctrl + N**」を押します。
3. **スクリプトコードをコピーする**：
   - URL を入力する。
   - https://drive.google.com/drive/u/0/folders/1ruQQpvJWnE9s8TC7uY18vZHzY1GOZx1P
4. **エディタに貼り付け**：
   - VS Code の新しいファイルにコピーしたコードを貼り付けます。
5. **ファイルを保存**：
   - メニューバーの「**File**」→「**Save As...**」をクリックします。
   - 保存場所を選択します（例：デスクトップ）。

- ファイル名を「**measurement.py**」と入力します。
- **重要**：拡張子が .py になっていることを確認してください。
- 「Save」ボタンをクリックします。

7.2. スクリプト内のパスやシート名の編集

スクリプト内で特定の部分を環境に合わせて編集する必要があります。以下の箇所を順に確認し、適切に変更してください。

1. **JSON ファイルのパスを編集**：
   - コンピューター上で JSON ファイルが保存されている正確なパスに変更します。

```
import serial
import gspread
import sys
from oauth2client.service_account import ServiceAccountCredentials
from datetime import datetime
import time
import threading

# Google Sheets APIの認証情報設定
scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']
credentials = ServiceAccountCredentials.from_json_keyfile_name(
    r'C:\Users\Administrator\Desktop\measurement\measurement-420905-6981cadefd35.json',
    scope
)
client = gspread.authorize(credentials)
```

2. **スプレッドシートの URL を変更**：
   - スプレッドシートを開き、ブラウザのアドレスバーから URL をコピーし、貼り付けます。

```
# スプレッドシートとシート名を指定
spreadsheet_url = "https://docs.google.com/spreadsheets/d/1cW70D35JDVfm90jVsdXC6rD6jyo1AVdT54DJt4-7vWM/edit?gid=810248108"
try:
    measurement_sheet = client.open_by_url(spreadsheet_url).worksheet("土耕栽培POST")  # 記録したいスプレッドシート名
except gspread.exceptions.WorksheetNotFound:
    print("エラー: 指定したシート '土耕栽培POST' が見つかりません。シート名を確認してください。")
    sys.exit(1)
```

3. **データを送信する POST を変更**：
   - この PC が送信する栽培方法の POST（名称完全一致）を入力。

```
# スプレッドシートとシート名を指定
spreadsheet_url = "https://docs.google.com/spreadsheets/d/1cW70D35JDVfm90jVsdXC6rD6jyo1AVdT54DJt4-7vWM/edit?gid=810248108"
try:
    measurement_sheet = client.open_by_url(spreadsheet_url).worksheet("土耕栽培POST")  # 記録したいスプレッドシート名
except gspread.exceptions.WorksheetNotFound:
    print("エラー: 指定したシート '土耕栽培POST' が見つかりません。シート名を確認してください。")
    sys.exit(1)
```

＊栽培方法が３つの場合には PC が３つ必要。であり、それぞれこの場所のみ変更する。

4. **シリアルポートを確認（変更）する**：
   - 使用するシリアルポートを確認する。違う場合には変更する。

```
# シリアルポートの設定
try:
    ser = serial.Serial('COM3', 9600, timeout=1)  # 使用するシリアルポートを確認
except serial.SerialException as e:
```

5. スクリプトの再保存

- 編集が完了したら、VS Code のメニューバーで「**File**」→「**Save**」をクリックします。
- または、キーボードショートカット「**Ctrl + S**」を押して保存します。

---

8. Python スクリプトの実行

8.1. スクリプトの実行

1. **スクリプトを実行**：
    - シリアルポートにマイクロビットが接続されていることを確認し、実行する。

2. **データの自動記録**：
    - 一定数のデータ（例：各センサーから 300 個のデータ）が収集されると、平均値が Google スプレッドシートに自動的に追加されるようになっている。
3. **スプレッドシート（OO栽培 POST）に反映されているか確認**：
    - A 列（タイムスタンプ）、B 列（温度）、C 列（照度）、D 列（水分率）

---

9. トラブルシューティング

スクリプト実行時に問題が発生した場合の対処法を以下に示します。

9.1. シリアルポートのエラー

- **エラーメッセージ例**：
- Error opening port: could not open port 'COM3': FileNotFoundError(2, '指定されたファイルが見つかりません。')
- **対処法**：
    1. **シリアルポート番号の確認**：
        - デバイスマネージャーで接続されているデバイスの COM ポート番号を再確認します。
        - スクリプト内のポート番号（例：COM3）が正しいことを確認し、必要に応じて修正します。
    2. **デバイスの再接続**：
        - デバイスを一度抜き、再度接続してみてください。

3. **ドライバーの確認**：
    - デバイスに必要なドライバーが正しくインストールされているか確認します。

9.2. スプレッドシートにデータが追加されない

- **エラーメッセージ例**：
- An error occurred while writing to the spreadsheet: SpreadsheetNotFound
- **対処法**：
    1. **スプレッドシート URL の確認**：
        - スクリプト内の spreadsheet_url が正しいことを確認します。
    2. **サービスアカウントの権限確認**：
        - スプレッドシートの共有設定で、サービスアカウントのメールアドレスに**編集者権限**が付与されていることを確認します。
    3. **JSON キーのパス確認**：
        - スクリプト内の JSON キーのパスが正しいことを確認します。
    4. **Google API の有効化確認**：
        - Google Cloud Platform で Google Sheets API が有効化されていることを再確認します。

9.3. UnicodeDecodeError のエラー

- **エラーメッセージ例**：
- Decode error: 'utf-8' codec can't decode byte 0xff in position 0: invalid start byte
- **対処法**：
    1. **データ形式の確認**：
        - シリアルデバイスが正しいフォーマット（例：light:300¥ntemp:25.3¥nmoisture:45.2¥n）でデータを送信していることを確認します。
    2. **エンコーディングの変更**：
        - 必要に応じて、デコード方法を変更します。例として、latin1 エンコーディングを試す：
        - data = ser.read(ser.in_waiting).decode('latin1', errors='replace')

ご不明な点がございましたら、お気軽に質問してください。 皆さんがスムーズにデータ収集を開始できることを願っています。