土曜講座「プログラミング初級入門」を通じた理数探究への考察

新井原　博　嗣（理　　科）

［目次］

1. はじめに
2. Python
　2.1. Pythonを選んだ理由
　2.2. Pythonの体験
　2.3. Pythonによるソフト開発(WEBスクレイピング)
　2.4. Pythonによるソフト開発(英単語学習ソフト)
3. Raspberry Pi Pico
　3.1. Raspberry Piとは
　3.2. Raspberry Pi Picoとは
　3.3. プログラミング実践
4. 土曜講座「プログラミング初級入門」
　4.1. 土曜講座とは
　4.2. スプレッドシートを用いた座席表作成
　4.3. Raspberry Pi Picoの組み立てと起動
　4.4. Raspberry Pi Picoへのプログラミング
5. 終わりに

1. はじめに

高等学校学習指導要領（平成30年告示）の改定に伴い、60期のカリキュラムにおいて大幅な変更が行われた。本校においても、「探究の授業化」、「文理コースの廃止を伴った理コースの設置」など、単なる教科内容の修正にとどまらない変更がもたらされた。文理コースと理コースの主要な違いをまとめる(表1)。

表1　文理コースと理コースの違い

	文理コース(59期まで)	理コース(60期以降)
1.コンセプト	文系にも理系にも進学できる	(原則として)理系進学する
2.授業内容	理系進学者向け	理系進学者向け
3.内部進学先	すべての学部	理工、商、経、国情、国営
4.その他	卒業論文	理数探究

中央大学に進学する上で多くの生徒が希望する法学部への進学ができなくなることで、理コース選択者は、従来の文理コースを選択する生徒に比べて70%程度(約50人)に落ち込むことを予測していた。しかし、実際には、60期生の76名が進学を希望する結果となった。

元々理系進学者向けの授業編成であり、理系教科では物理が4単位から5単位に1増えた。加えて、従来は国語科が実施していた「卒業論文」を、理科・数学科(情報科含む)が実施すつ「理数探究」へ変更した。学習指導要領では理数探究は理数科の教科として扱われていたものの、今回の学習指導要領の変更に伴い、普通科での実施が可能となった。スーパーサイエンスハイスクール(SSH)に認定されていない学校が実施するのは珍しいが、カリキュラムの検討の結果、導入に至った。

「プログラミング初級入門」は、理数探究の全体実施に向けた先行調査・実施のために開発した授業プログラムになる。元々は、本校の家本　繁元教諭(現中央大学理工学部准教授)が実践されていた講座だが、エッセンスだけでも

引き継げればと考え、2023年度に実施した。

2. Python

2.1. Pythonを選んだ理由

元々、自らは学生時代に教科として習ったこともなく、情報科の教諭でもないため、プログラミングの系統的な知識は持ち合わせていなかった。類する知識で言えば、excelの関数を使うぐらいで、マクロに関してもほぼ自分でプログラミングすることはできないレベルだった。

しかし、コロナ禍において、令和2年に学校が一斉に閉鎖されることになった際、経済産業省主導で「巣ごもりDXステップ講座」[1] という企画が実施されることを知り、そこで初めてプログラミングに触れることとなった。プログラミング言語として選んだのは、Pythonである。現在であればプログラミングと言えば、ChatGPTに代表される「生成AI」が主流となっているものの、2020年当時はその前段階とも言える「機械学習」であった。その機械学習において使われていた言語がPythonであったため、とりあえずPythonに取り組んだ。

2.2. Pythonの体験

「巣ごもりDXステップ講座」では無料で複数の講座を受講し、自らの適性を見極めるところから入った。学校は全国一斉臨時休業(令和2年2月28日～)に入り、生徒と同様に我々教員も自宅待機を余儀なくされていたため、時間的余裕が十分にあった。

講座の中で最も役に立ったのはスキルアップAI株式会社の「機械学習のためのPython入門講座」であった。これは動画が8日(1時間/日)分あり、

---

[1] 「巣ごもりDXステップ講座」…オンライン上でプログラミングの授業等を受けられる講座。登録制で無料のものもあった。現在ではマナビDX(https://manabi-dx.ipa.go.jp/)に変更されている

Anaconda prompt(Spiderを含む)と呼ばれるPythonを動かす環境の構築[2]から、一通りの機械学習に関する流れを学んだ。動画主導でも、環境構築から一通りの流れを体験できるのは初学者にとって重要だと気付かされた。

2.3. Pythonによるソフト開発①(WEBスクレイピング)

この時点では、書籍・ブログ等に書かれているPythonのコードを動かすことはできたが、自らコードを作成することはできなかった。それでも初心者向けのコードに複数取り組み、徐々にできることを増やしていった。初めて自分でプログラミングを通じて達成できたと感じたのは、「WEBスクレイピング」である。「WEBスクレイピング」とは、インターネット上のWebサイトから特定の情報を自動的に抽出する技術で、Webサイトやデータベースから特定のデータのみを抽出させることができる。家本元教諭より、プログラミングの勉強のために借りていた日経ソフトウエア[3]の中に解説があった。

この技術を用いる対象に目をつけたのは、「電車の遅延状況の把握」である。「やりたいこと」を実践するのではなく、「やれること」から適用範囲を探した形である。本校では、電車の遅延の管理を交通機関が発行する「遅延証明書」ではなく、生徒からの申し出に合わせてWEBで確認する方式にしている。。実際の作業では教務部の担当教員が毎日指定の時間に遅延情報がまとめられているサイト[4]を開き、印刷して、大黒板に掲示している。担任は生徒の申し出に応じて、掲示物を確認して管理している。しかし、生徒対応などの朝の雑務に追われることで抜け漏れが生じてしまうこともあるため、担当教員の精神的負担が大きい作業であった。

すでに環境構築しており、Pythonのコードも入手できたため、容易に実用化できた。記事の想定は不明だが、このプログラミングを通じコードを０から

---

[2] 中部大学総合情報センター教育支援
(https://edu.isc.chubu.ac.jp/hsuzuki/iip/anaconda/install1.html) が詳しい

[3] 日経ソフトウエア2020年1月号第2付録「Pythonが5日でわかる本　AI基礎編」(日経BP)

[4] Yahoo!Japan路線情報(https://transit.yahoo.co.jp/diainfo/area/4)を使用している

理解するのではなく、必要な部分だけを書き換えていく形が、プログラミングを学ぶ上で重要な要素であると考えるようになった。

コード1　WEBスクレイピングのpythonコード

```
import urllib.request
import datetime

res = urllib.request.urlopen('https://transit.yahoo.co.jp/diainfo/area/4')
html = res.read().decode('utf-8')
res.close()

now = datetime.datetime.now()
filename = 'train' + now.strftime('%Y%m%d_%H%M%S') + '.html'
f = open(filename,'w',encoding='utf-8')
f.write(html)
f.close()
print('train.htmlに保存しました')
```

実際に私が記事のコードから、灰色をつけたURLの部分だけ変更した(コード1)。このレベルのコードであれば、現在のプログラミングの知識をもってすれば一つ一つのコードの意味を理解し、自分で組み上げることは可能である。

しかし、一から組み上げることに学習の時間を費やすよりも、様々な小さい実践を積み上げることの方が、学習効果が高いと自分の体験の中では感じた。しかし、記事のコードにおいてURL部分を変更しただけでは問題点も多くあった。気がついた点を挙げておく。

① コードを動かすための環境を用意しておく必要がある　(ハードル：高)

② コードを指定の時間に手動で実行する必要がある　(ハードル：高)

③　コードを実行するとファイル名が同じになり、上書きされてしまう

（ハードル：低）

コード１は上記①～③のうち、③を解決するように改変している。具体的には下から5行目の「+ now.strftime('%Y%m%d_%H%M%S')」を付け加えた。trainというファイル名に「年・月・日・時・分・秒」を書き加えるという命令である。これが理解できれば、Yは年を表し、mは月を表しているということに生徒でも気がつくであろう。Python上で日時を加えるコードも自分で考えたのではなく、インターネット上で類似のことをやっている記事を探し、コードを流用した。現在であれば、OpenAIのChatGPTや、GoogleのBardなど生成AIを用いればもっと技術的なハードルが低くなる。なお後方の(　)内のハードルは実践した上での最終的な感想である。最終的にはすべての問題をクリアする環境づくりも実践したが、本題と外れるため割愛する。

2.4. Pythonによるソフト開発②(英単語学習ソフト)

Pythonを用いたソフト開発において、次に取り組んだのは「英単語学習ソフト」の作成であった。当時、担任をしていた生徒の中に、英単語テストでいつも合格点に到達できない生徒がいた。実際にその生徒に使うようにはしなかったが、この生徒が活用することを想定して開発した。

前回と同様に自分で一から開発することは難しいので、インターネット上で同様のことを実践している例を探した。この中で、中心的なアイデアの元になったのは以下の2つである。

①「tkinterを使って英単語学習アプリを作成しよう。」(ジョイタスネット)[5]

②「Pythonのtkinterで簡易タイピングゲームを作ってみた」[6]

①の時点で最低限必要な要素はクリアしたが、問題が固定されているため、定期的に行われる小テストのことを考えると実用的ではなかった。英単語学習

[5] https://joytas.net/programming/python/tkinter-animal-word
[6] https://zenn.dev/takahashi_m/articles/a272cea4c3c4d7bb29e5

ソフトという点において、②も同様に問題が更新できないという問題点があったが、より学習アプリとしてのイメージはこちらの方が近いものであった。しかし、この時点で大きな2つの問題に直面することになった。

[1]「tkinterが自分には扱いづらかった」

[2]「Pythonにおけるclassという概念がまだ理解できていない」

現在では、おそらく[1]に関してはクリアできるが、[2]は基本中の基本でありながら、まだおぼろげにしか理解できていない。プログラミング技術を伸ばすという観点では、[2]を優先して取り組むべきであると考えている。ただ、この後、プログラミングをソフト開発ではなく、Raspberry Piを動かす方向にシフトチェンジしたため、未だ本格的には取り組んでいない。

最終的に一通り開発したもののコード(コード2)と画面(図1)を、以下にまとめておく。

コード2　英単語学習ソフトのpythonコード

```
import PySimpleGUI as sg
sg.theme('LightGray1')
import csv
import pygame as pg
pg.init()

#問題データをリスト化
with open('word2.csv',encoding='utf8') as f:
    reader = csv.reader(f)
    line = [row for row in reader]

#問題データの問題数をカウント
with open('word2.csv',encoding='utf8') as f:
```

```
    row_count = sum(1 for row in f)-1

#初期設定
font = ("meiryo",15)
question_num = 1
questtion_jpn = line[question_num-1][1]
questtion_en = line[question_num-1][3]
questtion_sol = line[question_num-1][4]

#アプリの表示設定
layout = [[sg.Text('問題番号'),
 sg.InputText(font=font,default_text="1",key='form1')],
 [sg.Text('和　　訳'),
 sg.InputText(font=font,default_text=questtion_jpn,key='form2')],
 [sg.Text('英　　訳'),
 sg.InputText(font=font,default_text=questtion_en,key='form3')],
 [sg.Text('上記の[　]内に入る適当な語を下に入力し、解答確認ボタンを押し
て下さい')], [sg.Text('入力　　'), sg.Input(font=font,key='form4')],
 [sg.Button('解答確認'), sg.Button('クリア'), sg.Button('終了'), ¥
 sg.Text('残り問題数'),
 sg.Input(default_text=row_count, size=5,key='form5')]]
window = sg.Window('音単対策アプリ', layout)
#入力内容の正誤判定
while True:
    event, value = window.read()
    if event == sg.WIN_CLOSED or event == '終了':
        break
```

```
    if event == '解答確認':
        if value["form4"] == questtion_sol:
            pg.mixer.Sound("seikai.wav").play()
            sg.popup("正解です。次に進んで下さい。")
            question_num += 1
            row_count -= 1
            questtion_jpn = line[question_num-1][1]
            questtion_en = line[question_num-1][3]
            questtion_sol = line[question_num-1][4]
            window["form1"].update(question_num)
            window["form2"].update(questtion_jpn)
            window["form3"].update(questtion_en)
            window["form4"].update("")
            window["form5"].update(row_count)
        else:
            pg.mixer.Sound("huseikai.wav").play()
            sg.popup("不正解です。再度、入力して下さい。")
            window["form4"].update("")
    if event == 'クリア':  # clear keys if clear button
        window["form4"].update("")

window.close()
```

[注意点]

・questionのスペルが一部間違っています

・あらかじめ、モジュールとして「PySimpleGUI」「Pygame」のインストールが必要

・実際のソフトを試したい場合、データは以下のgoogleドライブよりダウン

ロードして下さい

＊学内関係者のみ自由にダウンロード可能

＊学外の方はメール（araihara@tamacc.chuo-u.ac.jp）にご相談下さい

https://drive.google.com/drive/folders/1n3j5jr17xm4MCZ2XzkXSS1_plrHZu5A1?usp=drive_link

図1　開発した英単語学習ソフト

①や②を参考と言いながら、コード2とにはほぼ原型は残っていない。構成要素として何が必要であるかを参考としたと考える程度である。その上で、課題・解決目標として次のA～Eを挙げた。その後、解決した方法は矢印→の右側にまとめておく。

A：問題データが固定されている　→　csvデータによる更新

B：tkinter(GUI)が難しすぎる　→　PySimpleGUIの導入

C：どこが解答欄か分からない　→　PySimpleGUIの導入

D：「正解」と表示が残ってしまう　→　PySimpleGUIの導入

E：音がないから寂しい　→　Pygameの導入

＊csvデータによる更新…　「#問題データをリスト化」部分

＊PySimpleGUI[7]の導入…「#アプリの表示設定」部分

＊Pygameの導入…　「#入力内容の正誤判定」部分

[7] https://www.youtube.com/watch?v=hdjDYGXbVws が詳しい

理数探究の中におけるプログラミングでここまで生徒ができたら、かなりの上位層になるであろう。実際の所要時間としては2022年の夏時点で30時間程度の開発時間をとっている。

3. Raspberry Pi Pico

3.1. Raspberry Piとは

Pythonによるソフト開発に一定の目処が立ったが、実際に自ら経験してみると、情報工学等を目指す生徒にとっては有用だが、専門外の生徒にとってはハードルが高い取り組みになると感じた。そのため、次にライトなプログラミングを検討した。

そこで、白羽の矢が立ったのが「Raspberry Pi」である。ラズベリーパイ(通称、ラズパイ)と読み、イギリスで教育用コンピューターとして開発された経緯から非常に低価格(2万以下)でありながら、通常のパソコンのようにインターネットに接続し、各種センサーを取り付け、プログラミングも実行可能であった。手始めにキーボードと一体化したモデルである「Raspberry Pi 400」を購入した(図2)。この時期はコロナ禍もあり、自宅で籠もって研究を続けられるため、ラズパイ関連は秋葉原などの専門店であってもほぼ入荷がない状況だった。ほぼ唯一と言って良い入手可能なモデルの「Raspberry Pi 400」で、試行錯誤することにした。

図2　Raspberry Pi 400(https://www.raspberrypi.com/より転用)

① OS(Raspberry Pi OS)の導入

…各種ソフトを動かすOS(Windowsのような基本ソフト)をインストールする必要がある

② 日本語IMEの導入

…キーボードの入力した内容を、文字に変換するためのソフト(IME)も必要

③ 無線接続の有効化

…無線の部品は内蔵されているが、設定をして有効化する必要がある

④ センサーの物理的接続

…「Raspberry Pi 400」とセンサー等を接続するためには、端子の違いを理解する必要がある

⑤ センサーの動作

…センサー[8]を物理的接続した後、Pythonを用いて動作させる必要がある

(センサーメーカーが動作確認用のプログラムを無料で公表している)

これでは不十分ではないかと当時は考えていたが、今、思い返せば生徒は①～⑤を実践するだけでも十分な取り組みである。「Raspberry Pi 400」では学校で配布されていたUSBカメラを接続して連続撮影し、LEDをプログラムに応じて点滅させた。ただ、「Raspberry Pi 400」が低価格とは言え2万円程度かかるとすると、学校が「貸し出す」という方式をとるしかない。そこで、生徒が購入しても負担感が少ない3,000～5,000円というレベルでできないかと目をつけたのが、次の「Raspberry Pi Pico」である。

3.2. Raspberry Pi Picoとは

Raspberry Pi Pico(図3)は一般的にはラズパイピコと読み、マイコンに分類される。ラズパイピコは自分自身だけではアプリケーションを制御、操作することはできない。あらかじめ記述したプログラムのみを記録、実行すること

[8] センサーキットとしてSunFounder(https://www.amazon.co.jp/dp/B09BMSS41D)等を使用

で、GPIOに接続された他のデバイスとのやり取りを行う。そのため、ラズパイよりもやや上級者向けであるが、低価格・低消費電力など魅力も大きい。

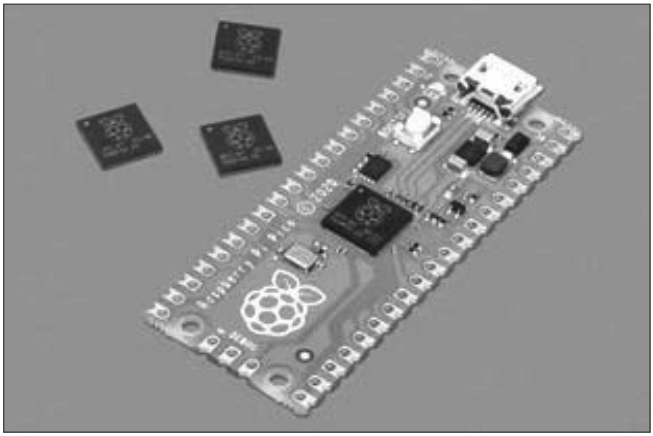

図3　Raspberry Pi Pico(https://www.raspberrypi.com/より転用)

表2　ラズパイとラズパイピコの違い

	ラズパイ(ラズパイ400等)	ラズパイピコ
1.分　　類	パソコン	マイコン
2.価　　格	1～2万円程度	1,000円前後
3.主な用途	パソコン	電子工作
4.プログラミング	可能	必須

2023年7月から、ラズパイピコを用いた電子工作に取り組み始めた。秋葉原の電気街には自身が大学生の時代から行っていたものの、本格的な電子工作のために訪れたことはなかった。秋葉原において電子工作で有名なお店は秋月電子や千石電商[9]である。これらのお店ではパーツ1つから購入することができ

[9] 秋月電子(https://akizukidenshi.com/catalog/)、
千石電商(https://www.sengoku.co.jp/)

るため、購入する物品が明確になるレベルまでスキルが上達してくれば非常に有用な場所となる。

3.3. プログラミング実践

ラズパイ400での経験もあるため、この時点ではHP上で公開されているソゾラブログ[10]などの記事を読めば、機材の購入は問題なく行え、初心者向けのプログラミングであれば多少改変できるレベルに到達している。このレベルに生徒が到達できるように準備ができれば、あとは生徒が自由にコントロールできるようになると考えた。土曜講座の「プログラミング初級入門」ではこれを最終的な到達点に定めた。以下、ラズパイピコを用いて実践した電子工作を列挙しておく。

① DHT20・DHT11(温度・湿度センサー)を用いた電子工作
② DS18B20(温度測定プローブ)を用いた電子工作
③ HC-SR04(有機ELミニディスプレイ)を用いた電子工作
④ BME280(温度・湿度・圧力センサー)を用いた電子工作
⑤ DHT20で測定した温度・湿度をHC-SR04に表示する電子工作

①～④は様々なサイトで解説されているため、生徒がその解説を理解できるレベルに到達していれば、試行錯誤しながらでも取り組むことは可能である。しかし、工夫のレベルを上げようとするならば、⑤が目標となるであろう。⑤は①と③を組み合わせたものになるが、電子工作で一般的な部材であるDHT20とHR-SR04であっても、組み合わせるところまでいくとインターネット上でも解説はほとんどされていない。理数探究が始まって3～5年程度でここまで到達できる生徒が現れることが理想的である。

土曜講座では①を3時間(1時間×3コマ)で実践することを目標とした。詳しいプログラミングのコードなどは後ほど記載するが、生徒が取り組むには予想

[10] Raspberry Piについて詳しいブログ(https://sozorablog.com/)

以上の準備が必要で、こちらが準備したコードをラズパイピコに組み込み、改変の説明をするまでが限界であった。

4. 土曜講座「プログラミング初級入門」

4.1. 土曜講座とは

土曜講座は、本校で行われている課外授業の1つで、学年やクラスを問わず、興味をもった生徒が参加する授業である。59期の生徒までは総合学習の単位取得目的で受講している場合があったが、60期の生徒からは総合的な探究の時間の単位とは切り離されることになったため、完全に興味・関心に沿って受講している。

[プログラミング初級入門の概要]

①場所：59期(3年)はPC端末をもっていないため、Windows端末のあるPC教室

②人数：24名(完全抽選、希望は30名程度だったと報告を受けている)

③担当：新井原のみ

④講義：分散した形で8回(準備期間をとり、軌道修正可能にするため)

前半：スプレッドシートを用いた座席表作成、後半：ラズパイピコのプログラミング

[1] プログラミングの考え方を学ぼう、UI(User Interface)の考え方

[2] googleフォーム・googleスプレッドシートの作り方、データの共有

[3] 関数を用いたデータの処理、座席表のアイデア

[4] データ処理、座席表のアイデアの実装

[5]「ランダム」「男女を交互にする」「学年ごとに色分け」を実装するアイデア

[6] Raspberry Piとは

[7] Raspberry PiとThonny

[8] Raspberry Piの組み立てとプログラミング

そもそも生徒のPC端末に関する知識は一定ではないことが想定されていたため、「スプレッドシートによる関数処理」→「ラズパイでのプログラミング」につなげることを考えた。全員がChromebookを使用している環境にもできなかったため、こちらの想定以上に一つ一つの操作説明に時間がかかる状況だった。8回のうち、2回は時間調整用に想定したがこれでも時間が超過するかもしれないという内容であった。

4.2. スプレッドシートを用いた座席表作成

googleスライドを用いて座席表のUIを考えた(図4)。右半分の正方形は生徒の使用した座席を表し、4色で塗られている。黄・赤・青は本校では学年を区別するための色(上履きや体操着に使用)として導入されている。「学年カラー」を座席表に反映させたいというアイデアである。

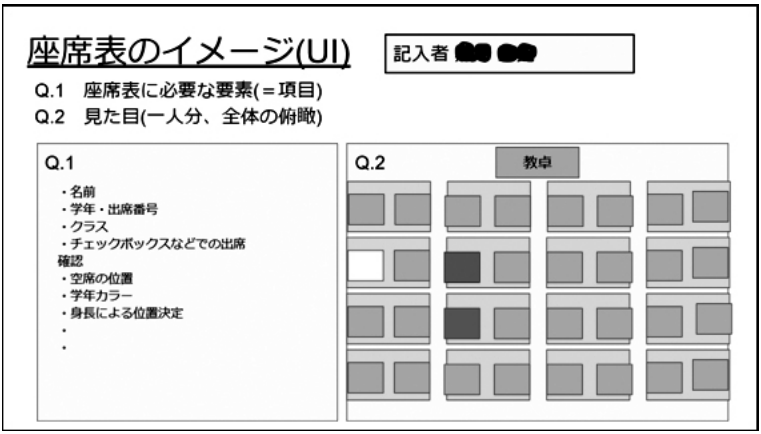

図4　生徒が作成した座席表のイメージ(UI)

次に座席表に用いるダミーデータを生徒と作った。ダミーデータという概念とともに、googleフォームの作り方を同時に教えた(図5)。実際に自分たちが作ったデータであるため、普段よく知らない同級生や異なる学年の先輩・後輩の特徴も分かり、アイスブレイクとしても有用だった。

図5(a)　googleフォームを用いたダミーデータの収集・共有

図5(b)　googleスプレッドシートに共有されたダミーデータ(一部加工)

作成したダミーデータをスプレッドシートに反映させれば、座席表は完成する。しかし、上記の図5(b)のような場合、「年・組・番」はスプレッドシート上で文字列扱いされていることに着目し、文字列を数字に変換する必要がある

ことを説明し、関数の処理方法を説明した(図6)。

図6　関数の処理方法

座席表は普段の生活で目にするものなので、どのような要素が必要かはイメージしやすい。ここでは、データベースと接続するために何が必要なのか説明した(図7)。

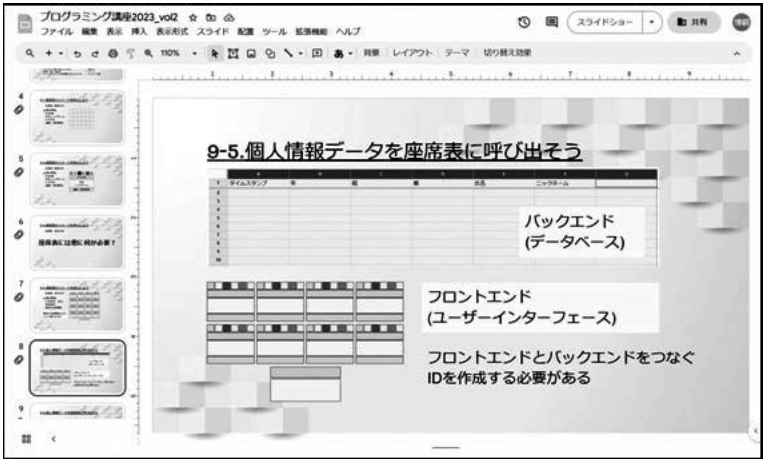

図7　UIとデータベースをつなぐ概念

この段階で別途どのような機能を実装すれば便利になるか生徒のアイデアを募集し、どのような解決方法があるかの一例を示した(図8)。

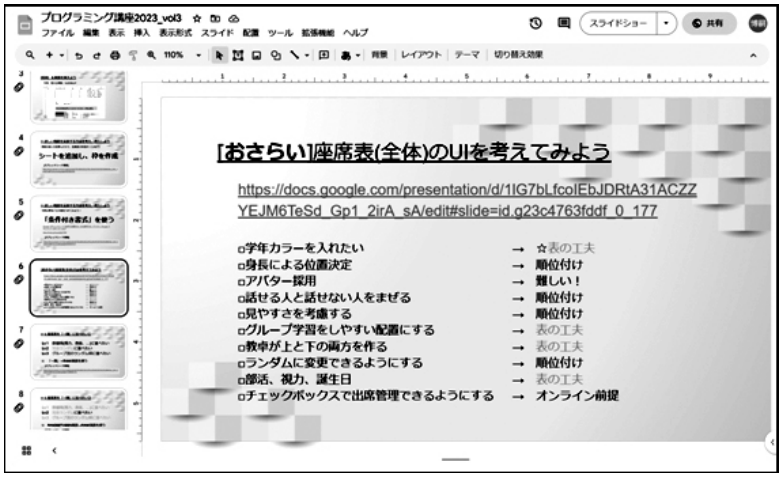

図8　生徒の実装したい機能と解決方法

実際の授業では「ランダム」と「学年カラー」を実装した。「ランダム」はRANDBETWEEN関数＋RANK関数により、「学年カラー」は、条件付き書式の手法により実装した(図9)。

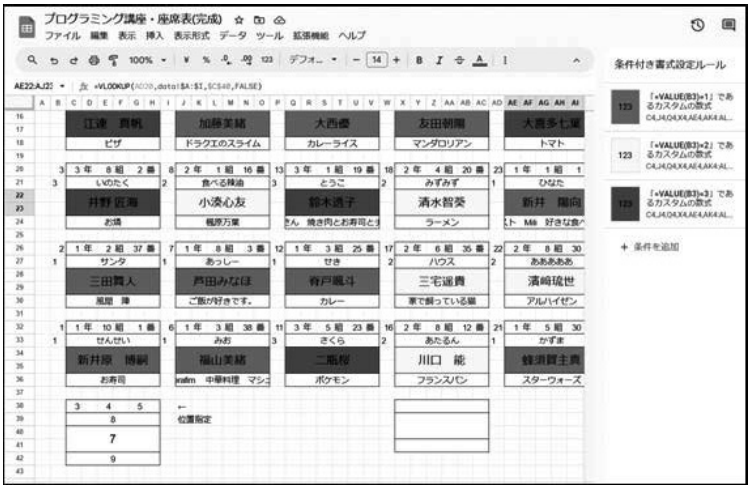

図9　条件付き書式による色分け

その上で、実現させた最も難しい例として、「グループ別ランダム」を解説した(図10)。

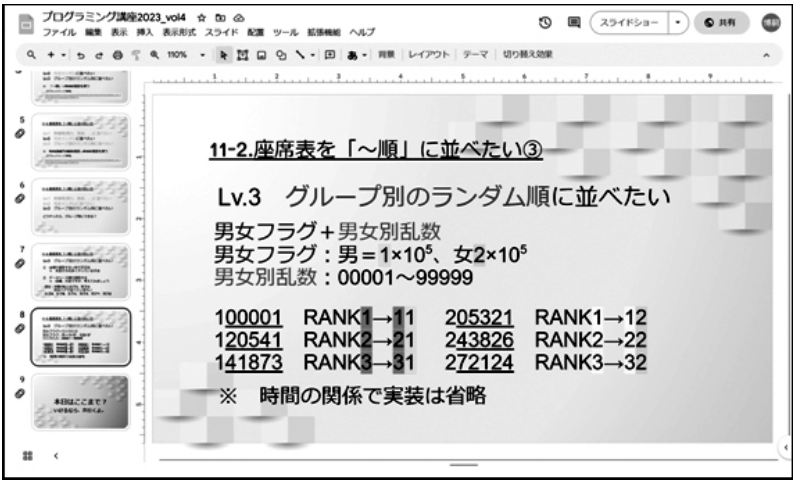

図10　グループ別ランダム

男女別を参考にフラグ立てをした上で、先程のRANDBETWEEN関数とRANK関数を組み合わせている(図11)。このように、一つのアイデアを拡張してより複雑なルール作りを実装できると生徒も能力の向上を意識しやすくなるため、学習効果が高まると考えた。

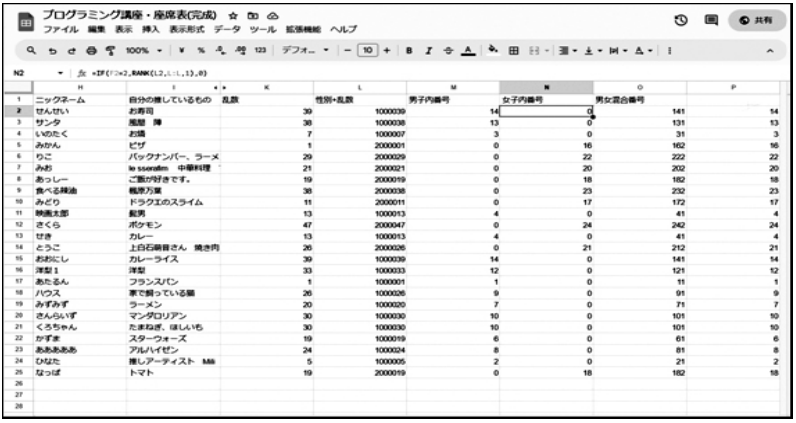

図11　男女別ランダム実装の処理

4.3. Raspberry Pi Picoの組み立てと起動

土曜講座において実際の組み立てを実施するのは2023年9月以降であったため、2023年の8月に準備を行った。自分自身で組み立てするのとは異なり、生徒自身がスライドを見れば作業が進められるようにするための準備が必要だった。3回分の授業内容であったが、想定を上回る時間がかかることを予測し、2回分の講義内容を準備した。

・電子部品(Raspberry Pi Pico含む)とプログラミング開発環境Thonnyの理解

・電子部品の組み立て、Thonnyの起動とプログラミング

この講座を成立させるためには、以下のものが必要であった(図12)。

① Windowsが起動するPC…PC端末の流用(ただし、データは再起動により破棄される)

② プログラミング開発ソフト
Thonny Portable(https://portableapps.com/node/62669)

③ microUSBケーブル(メスTypeA－オスTypeB)…100均のもので問題ない、私物流用

④ ラズパイピコ(Raspberry Pi Pico)…単価,1000円程度

⑤ ブレッドボード…単価300円程度

⑥ DHT20(温度・湿度センサー)…単価400円程度

⑦ ジャンパーワイヤー(配線用ケーブル)5本…単価10円程度

※ ③～⑦を9個(生徒使用分8個＋演示用1個)ずつ用意

図12　電子部品(Raspberry Pi Pico含む)

図13　プログラミング環境

※　Thonny Portableはインストール不要なので、学校用の共有PCでも利用可能

③～⑦はあらかじめ数を調整し、プラスチックケースに必要な分をすべて仕分けしておいた。毎回、同じものを使うことができるようにチーム(3人1チーム)を構成し、チーム名をプラスチックケースに記入して管理するようにした。24名の受講者のうち最終的には2名が脱落(詳細不明)した。担当教員一人がカバーできる人数と実効性を考慮すると、2名1チーム×8チーム＝16人が理想的だと考える。相談しながら実施するので、必ずしも1人1台は必要ない。

図14に示すように、壊れやすい端子を保護するため、ブレッドボードへの差し込みをまず行った。8班のうち1班は上下を逆にしていたが、途中で気が付き、修正した。

図14　ラズパイピコとブレッドボードへの差し込み

予想外に時間がかかったのは、プログラミング開発ソフトThonnyを起動する準備だった。PCの教室のネット環境が悪く、同時に同じサイトにアクセス

したためかダウンロード自体に時間を要した。また、Windows上でzipファイルを展開してファイルを起動するというごく単純な操作でも、口頭説明では理解できない生徒が沢山いたことが誤算だった。

図15　PCからラズパイピコへの接続

図15のように接続し、Thonnyで動かすためのuf2ファイルをインストールした(図16)。

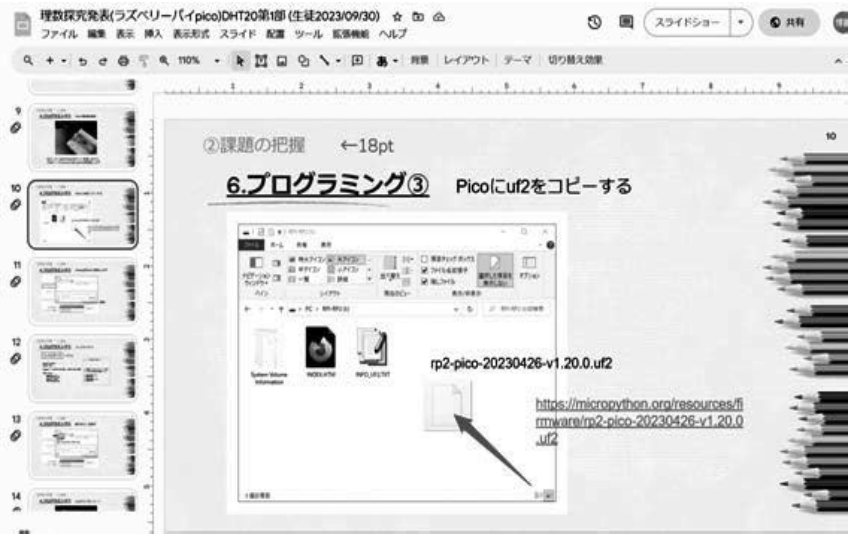

図16　uf2ファイルのインストール

本来はこの後に、LEDを点滅させるプログラムを動かす予定であったが、初日は断念した。

4.4. Raspberry Pi Picoへのプログラミング

2回目にはThonny Portableやuf2ファイルは、PC端末のデスクトップ上に準備しておく工夫をした。再度、図15や図16の作業を思い出しながら、Thonnyを起動させた。

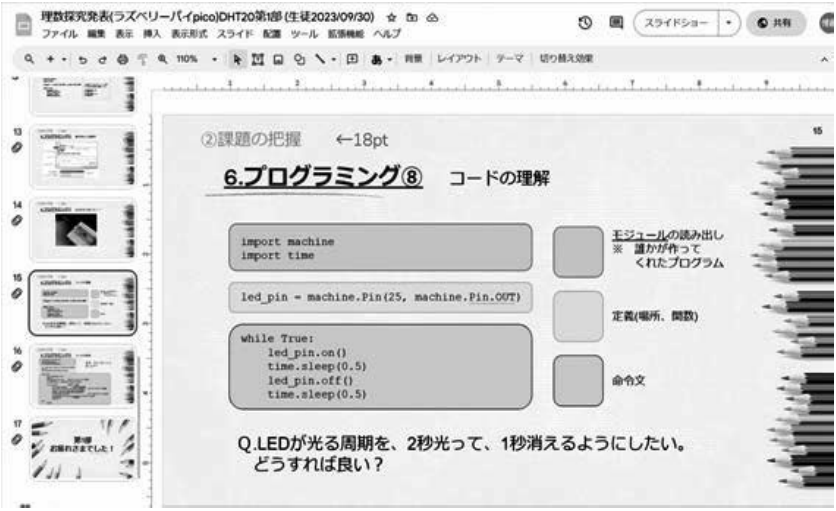

図17　コードの理解

講座後はすべての電子部品を回収していたため、2回目の講座前にLEDが光るコードはラズパイピコ内に準備し、プログラムを実行した。その上で、コード(コード3)の意味を考えさせた。

コード3　Lchika.py

```
import machine
import time

led_pin = machine.Pin(25, machine.Pin.OUT)

while True:
    led_pin.on()
    time.sleep(0.5)
```

```
    led_pin.off()
    time.sleep(0.5)
```

私自身もそうだが、生徒がすべてのコードを1から理解することを目指すのは難しい。そこで、コード3を動作させた後、「2秒光って1秒消える」ように加工する課題を出した。理解できない生徒には、コード3では「0.5秒光って0.5秒消える」ことをヒントとして出した。最終的にはコード4のように工夫すれば良いことに気が付き、実践していた。

コード4　Lchika_kai.py

```
import machine
import time

led_pin = machine.Pin(25, machine.Pin.OUT)

while True:
    led_pin.on()
    time.sleep(2)
    led_pin.off()
    time.sleep(1)
```

※　グレー部分が変更場所

この時点で2回目(通算7回目)の授業が終了した。3回目(最終8回目)は、コード3を実行するところまで生徒が自力で到達した。最終回では、ラズパイピコで温度・湿度センサーのDHT20を動作させることが目的だった。前回までの運営の反省を活かし、あらかじめプログラム自体はこちらでラズパイピコに準備しておいた(コード5)。センサーの説明および組み立てに関しては予想以上に早く組み立てることができた(図18)。

コード5　measure.py

```
from machine import I2C
from dht20 import DHT20

i2c = I2C(1)
dht20 = DHT20(i2c)

while True:
temper = dht20.dht20_temperature()
humidity = dht20.dht20_humidity()
print("temper : " + str(temper))
print("humidity : " + str(humidity))
```

図18　ラズパイピコとDHT20の接続

コード5を実行してみると分かるが、Thonnyのシェル部分(図19)に、立て続けに温度と湿度の計測されたものが表示される。そこで、最後の課題として、「1秒おきに温度と湿度を測定するようにする」を出した。実は、これは事前に準備しておいたものではなく、講座を実施している間に思いついた。コード3,4で時間制御を学んでいるので、それを応用すると、一番下の行(print("humidity：" + str(humidity)))の次に、「time.sleep(1)」とたった1行加えるだけで良い(コード6)。実際に生徒が実行してみると、表示間隔が1秒ごとになることが確認できた。生徒たちも学んだ内容が繋がり、歓声が上がった。

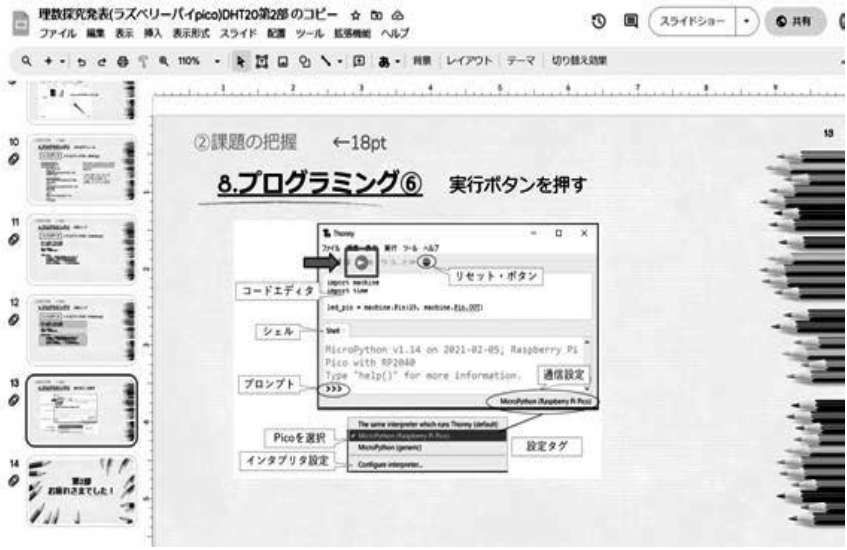

図19　Thonnyの画面(青い囲みの下半分がシェル)

コード6　measure_kai.py

```
from machine import I2C
from dht20 import DHT20
```

```
i2c = I2C(1)
dht20 = DHT20(i2c)

while True:
    temper = dht20.dht20_temperature()
    humidity = dht20.dht20_humidity()
    print("temper : " + str(temper))
    print("humidity : " + str(humidity))
    time.sleep(1)
```

5. 終わりに

スーパーサイエンスハイスクール(SSH)でもない学校にとって、「理数探究」は機材もノウハウも乏しく、教員にとっても難易度の高い教科である。しかし、従来の教科教育では伸ばすことが難しい[1]～[3]も、「理数探究」では実践を通して身につけることができると考えた。

[1]　実社会での生活に活きる力の育成

[2]　現代社会の課題を学ぶこと

[3]　周囲との関係の中で将来どのように生きるのかという自分の特性

[1]～[3]を意識する中で大事なことは、生徒が自らの興味・関心に沿った分野に挑戦し、限られた条件の中で成果を出すことにある。必ずしも生徒たちが全国大会で表彰されるような優れた結果を出すことではない。また、生徒だけでなく教員も自らの専門性を大事にしながら、自分なりのアプローチを示すことも必要である。

「課題が出されたものに取り組んでいく」授業形式は、本質的に受動的なものになりやすい。しかし、生徒自らがテーマを見つけることになると、自からの興味・関心と向き合わざるを得ない。最初はこれはやりたくないという否定的な考えばかりであったとしても、様々な経験を経るうちに、徐々に絞られ、

自らの興味・関心が明らかになってくる。

一見するとデメリットと捉えられがちな、「限られた条件」の中で取り組むことにも意味がある。社会人として解決すべき課題や問題があったとしても、多くの場合は複数の要素が絡み、唯一無二の完璧な解答などない。限られた条件の中でバランスをとり、最善を尽くすことが必要である。例えば生徒にとって施設面で限られたものがあっても、その中で工夫して(一定程度)解決したという成果こそ評価されるべき点である。

本稿ではプログラミングを中心とした「理数探究」について考察を行った。まだ、現時点ではプログラミング能力がすべての生徒にとって必要な能力なのかは分からない。サイボウズのkintone(キントーン)などに代表されるノンコードプログラミングが、将来的には主流になっていくかもしれない。しかし、自分で仕組みを理解し、試行錯誤しながらも自分の思い通りに動かす経験は、プログラミングの大きな魅力となりうる。この点において、理数探究の研究テーマの一つにプログラミングは最適だと考える。