

Vol. 101

CONTENTS

【コラム】プログラムを投稿してみませんか… 坂東 宏和

【解説】Processing でプログラミングに挑戦!—第1回 図形を描いてみよう—… 杉浦 学

【解説】第12回全国高等学校情報教育研究会全国大会 (和歌山大会) Next Stage… 肥田 真幸

COLUMN

プログラムを投稿してみませんか



中学生のときに MSX2+ というコンピュータを買ってもらい、そこからプログラマとしての道が始まりました。当時は、複数の雑誌に読者から投稿されたプログラムを掲載するコーナーがあり、そこに掲載されたゲームのプログラムを実行したり、改良（改悪？）したりして楽しんだことを憶えています。現在のようにプログラムをネットからダウンロードできるようなことはなく、紙に印刷されたプログラムを自分で入力（いわゆる写経）するしかありませんでしたので、入力ミスでゲームが止まるとか、無敵になるとか、いろいろと経験しました。写経プログラミングの学習効果については賛否があるようですが、少なくとも他人のプログラムを読むことには意味があるようで、プログラミングの基礎を自然に学ぶことができたように思います。その後、本格的なゲームを作りたいと思い、高校・大学と情報系の学科に進学、途中でゲーム作りを学ぶには別の学校だったんじゃないか？とやや後悔しつつも、今はゲームとは無縁のプログラムをそれなりに楽しく作っています。

さて、本誌では、本会セミナー推進委員会にご協力いただき、ジュニア会員の皆さんがプログラミングを始めるきっかけや目標になればと考え、連載「集まれ！ジュニア会員！！」の中で、皆さんから投稿されたプログラムを紹介しています。また「ぺた語義」では、今号から4号連続の予定で、プログラミング言語 Processing の入門記事を掲載します。

ジュニア会員の皆さん、開発したプログラムを本誌に投稿してみませんか？ Scratch や Processing はもちろん、その他、どのような言語で開発したプログラムでも OK、内容も自由です。今のところ投稿が少ない状況ですので、掲載率はかなり高めです。積極的な投稿をお待ちしています。

坂東宏和 (獨協医科大学)

「集まれ！ジュニア会員！！」の投稿方法

対象作品：オリジナルのプログラムであれば、プログラミング言語・内容はどのようなものでもかまいません。

投稿方法：(18歳未満の方は保護者の同意をもらってから) 下記の情報を電子メールで本会誌編集部 (editj@ipsj.or.jp)宛に送付してください。

- ・氏名、ニックネーム (掲載時の名前)、連絡先メールアドレス、(本会会員の場合には) 会員番号
- ・作品に利用しているプログラミング言語
- ・作品のタイトル、作品の説明とこだわったポイント (簡単で OK)
- ・プログラム一式 (メールの添付ファイルとして送付してください。Scratch のようにネット上でプログラムを確認できる場合には、URL だけでもかまいません)

その他：掲載が決まった際には、本会ジュニア会員になっていただく必要があります。また、本会による作品の無償公開をご承諾いただいた上で、承諾書等^{★1★2}を提出していただく場合があります。

掲載された方には、掲載誌、および、IPJSJグッズを差し上げます。

★1 論文付録データの取り扱いに関する規程 (<https://www.ipsj.or.jp/copyright/ronbun/supple.html>)

★2 論文付録データの学会利用に関する承諾書・チェックリスト (https://www.ipsj.or.jp/copyright/ronbun/furoku-shodakusho_checklisti.html)

Processing でプログラミングに挑戦！

—第1回 図形を描いてみよう—

杉浦 学

鎌倉女子大学

この連載について

この連載では、プログラミングの経験が少ないジュニア会員を対象に、プログラミングの入門記事を4回にわたって掲載していきます。命令をマウスで組み立てる (Scratch のような) タイプのプログラミング経験があり、命令を文字で入力することに挑戦したいというジュニア会員の皆様 (主に中学生～高校生) を読者として想定しました。

記事の中には「練習問題」や次号までの「宿題」を用意してあります。文章を眺めるだけでなく、実際に手と頭を動かしながら読んでみてください。

Processing とは

プログラミングを簡単に説明すれば「コンピュータが理解できる言葉を使って、実行してほしいことをコンピュータに伝えること」といえるでしょう。

この連載では、コンピュータが理解できる言葉として、「Processing (プロセッシング)」を利用します。Processing を使えば、画面に図形を表示して動かしたり、ユーザの操作に反応したりする仕組みを手軽に作ることができます。

Processing は Java というプログラミング言語をもとにして作られています。短い命令でいろいろなことができるように工夫されており、プログラミングの初心者でも扱いが簡単です。Processing の生い立ちなどを詳しく知りたい場合は、文献1) の『Processing をはじめよう 第2版』を読んでみてください。

準備

さっそく Processing を使ったプログラミングの準備をはじめましょう。Processing の公式サイトのダウンロードページ (<https://processing.org/download/>) にアクセスし、お使いの PC のオペレーティングシステム (OS) にあったファイルをダウンロードしてください (図-1)。Windows で使う場合の手順を簡単に解説していきます。

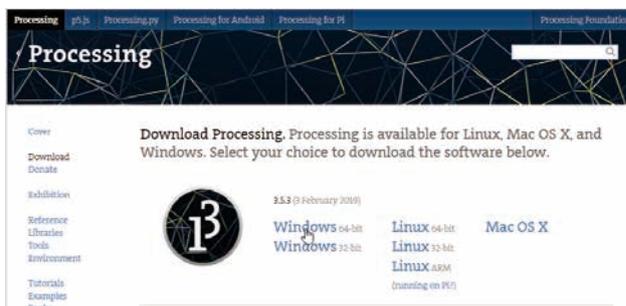
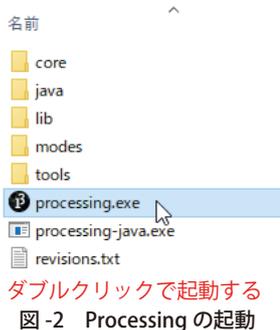


図-1 OS にあったファイルをダウンロード

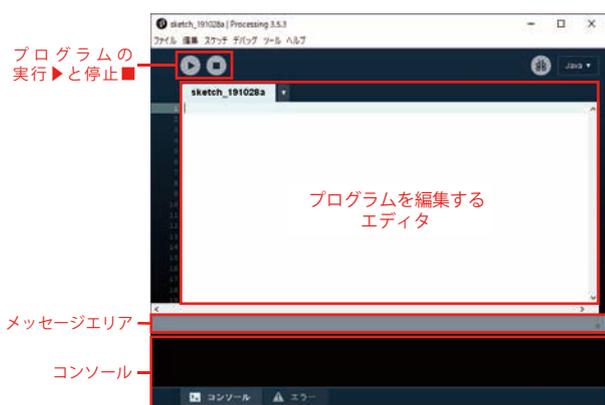
Windows の場合は「processing-X.X.X-windows64.zip」という圧縮ファイル (X.X.X はバージョン番号) がダウンロードされるので、これを展開します。展開が完了すると「processing-X.X.X」というフォルダができるので、分かりやすいところ (たとえばデスクトップなど) に移動しておきましょう。フォルダの中の「processing.exe」をダブルクリックすれば、Processing が起動します (図-2)。設定によってはセキュリティの警告が表示される場合がありますが、その場合は「アクセスを許可する」を選択してください。



初回の起動時には「Welcome to Processing 3」と書かれた小さなウィンドウが表示されます。「Show this message on start up」のチェックを外してから、「Get Started」をクリックします(図-3)。

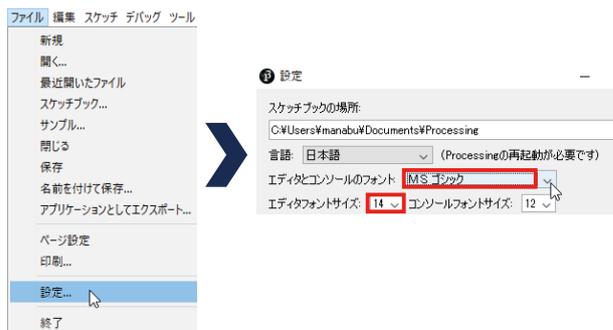


Processingを起動した直後の画面を図-4に示します。



最初に日本語の入力と表示ができるように設定しておきましょう。ファイルメニューから「設定」をクリックし、「エディタとコンソールのフォント」を日本語が表示できるものに変更しておきます(図-5)。エディタのフォントサイズも少し大きめ(たとえば

14)にしておくとういでしょう。画面の例では「MSゴシック」の「14」ポイントに設定しています。



さっそくプログラムを書いて、実行してみましょう。Processingではプログラムのことを「スケッチ(Sketch)」と呼びます。

最初は円を描いてみましょう。スケッチ1の内容をエディタにすべて半角で打ち込んでください。入力ができたら、プログラムを実行するために▶のボタンをクリックし、結果を確認しましょう。

```
ellipse(50, 50, 80, 80);
```

スケッチ1 最初のスケッチ(円を描く)

小さなウィンドウが開き、そこに円が表示されず(図-6)。スケッチの実行を停止するには■のボタンをクリックします。命令の意味については、あとで詳しく説明していきます。エラーが表示されてうまく動作しないときは、スケッチ1の内容が間違いなく半角で入力できているかを確認してください。



新しいスケッチには「sketch_XXXXXXX」という名前が自動的につきます。XXXXXXXは6桁の日付とaからはじまるアルファベットです。スケッチを保存するには、ファイルメニューから行います。Windowsの場合は、ドキュメントのProcessingというフォルダの中に、スケッチごとにフォルダが作成されて保存されます。たとえば、2019年の10月28日に最初に作ったスケッチは「sketch_191028a」というフォルダの中に保存されます。フォルダの中にある「sketch_191028a.pde」というファイルをダブルクリックすると、編集の続きが行えます。スケッチの名前は自由に変更することができますが、全角の文字は自動で「_（アンダーバー）」に置換されます。半角で分かりやすい名前を入力して保存するようにしましょう。

基本的な命令の形

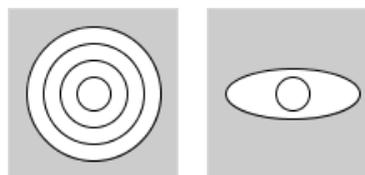
最初に試したスケッチ1の内容を詳しく解説し、命令の基本的な形やエラーについて整理しておきます。

スケッチ1は「左から50ピクセル、上から50ピクセルの位置を中心にして、幅80ピクセル、高さ80ピクセルの円を描け」という命令です。人間が相手の場合、紙を渡して「円を描いて」と頼めば、ほどよい大きさの円を描いてくれると思いますが、コンピュータの場合はそうはいきません。どの位置にどのような大きさの円を描くのか、細かく指示してあげる必要があります。

□ 練習問題

スケッチ1に円を描く命令を追加して、お手本を参考的にと目玉の模様を描いてみましょう。

ヒント：上に書いた命令から先に実行され、画面の奥から図形が描かれます。



<解説>

的の模様を描くスケッチの例

```
ellipse(50,50,80,80);  
ellipse(50,50,60,60);  
ellipse(50,50,40,40);  
ellipse(50,50,20,20);
```

目玉の模様の解答例は、スケッチ3を参照してください。

命令の基本形を図-7に整理しました。これを理解すれば、ほかの図形の描画などにも応用できます。Processingにはさまざまな「関数」と呼ばれる部品が用意されています。それらのうちの1つが円を描くためのellipse関数です。関数は「よく使う便利な命令の集まりに、分かりやすい名前をつけて使えるようにしたもの」と理解すればよいでしょう。関数の処理に必要なパラメータ（動作を決める数値や文字など）をカッコで囲んで指定します。円の場合は4つのパラメータを指定して、描きたい円の詳細をコンピュータに伝えます。

先頭に関数の名前を書く パラメータはカッコで囲む 最後はセミicolon

```
ellipse(50,50,80,80);
```

複数のパラメータはカンマで区切って指定（円の場合は4つ）

※参考

命令のブロックを文字で作っていると考えるとよい

円を描く x: 50 y: 50 幅: 80 高さ: 80

図-7 命令の基本的な形

命令を書く際には、大文字小文字も含めて厳密に一字一句間違えないようにする必要がありますが、打ち間違いや勘違いは誰にでもあるものです。入力をしている途中や、スケッチを実行しようと思ったときに赤い波線が表示されたり、行が黄色でハイライトされたりすることがあります。これはスケッチの文法間違いなどのエラーを示しています。エディタの下のほうにあるメッセージエリアに、エラーの内容が表示されます。それをヒントに修正をする必要があります(図-8)。



図-8 エラーの表示

□ 練習問題

次のスケッチはエラーで実行できません。修正すべき部分を考えてみましょう。実際にエディタに入力してみるのもよいでしょう。

```
ellipse 50,50,100,100;
ellipse(50 50,80,80);
elipse(50,50,60,60)
```

<解説>

- 1行目：パラメータを囲むカッコがない
- 2行目：最初の50の後にカンマが抜けている
- 3行目：ellipseのつづりが間違っているし、行末のセミコロンがない

スケッチの命令はすべて半角で入力します。誌面だと分かりにくいですが、スケッチ2の1行目は

最後のセミコロンが全角です。2行目は2番目のパラメータの前に全角のスペースが入っています。スケッチ2のように書いてしまうと、エラーが発生して実行ができません。

```
ellipse(50,50,80,80);
ellipse(50, 50,80,80);
```

スケッチ2 全角の入力でエラーになる例

命令の本体を書くのに全角は使えませんが、スケッチに日本語でメモ書きを加えることはできます。これを「コメント」と呼びます。複数行の場合は / * と / * で囲み、1行の場合は // をはじめの部分に記入します。コメントは灰色の文字色で表示されます。目玉の模様を描くスケッチにコメントを追加した例がスケッチ3です。

```
ellipse(50,50,80,30); //白目を描く
ellipse(50,50,20,20); //黒目を描く
```

スケッチ3 目玉の模様を描く(コメント付)

Processingのエディタにはスケッチを自動的に読みやすくしてくれる機能があります。編集メニューの「自動フォーマット」がそれです(図-9)。こまめに自動フォーマットをすれば、カンマの後にスペースが挿入されたりして、スケッチを読みやすい状態に保つことができます。



図-9 自動フォーマット



座標と図形

コンピュータの画面はピクセルと呼ばれる細かい点が集まってできています。この点に座標をつけることで、画面上の位置を指定します。Processingの場合は、実行結果を表示する画面の左上が原点(0, 0)に設定されます。画面の大きさを指定しない場合は、自動的に縦横が100ピクセルの画面が開きます。スケッチ1を実行したときの画面の大きさと座標を図-10に示します。

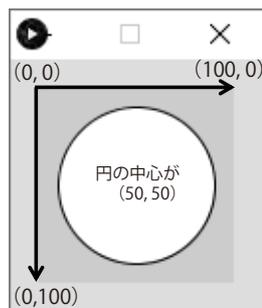


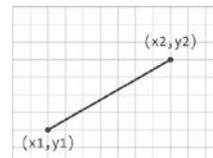
図-10 画面の大きさと座標

画面の大きさを指定したい場合は、スケッチの最初に size 関数による設定を加えます。幅480、高さ120のウィンドウを開き、ウィンドウの中心(240, 60)の座標に1ピクセルの大きさの点(point)を描くには、スケッチ4のように命令します。

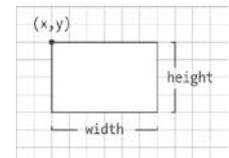
```
size(480,120);
point(240,60);
```

スケッチ4 ウィンドウの中心に点を描く

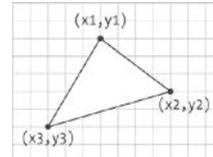
基本的な図形を描画するための関数と座標などのパラメータを図-11に整理しました。試しにいろいろな図形を描いてみましょう。



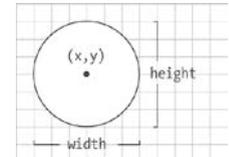
line(x1, y1, x2, y2)



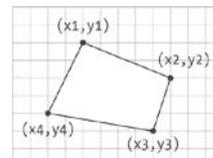
rect(x, y, width, height)



triangle(x1, y1, x2, y2, x3, y3)



ellipse(x, y, width, height)

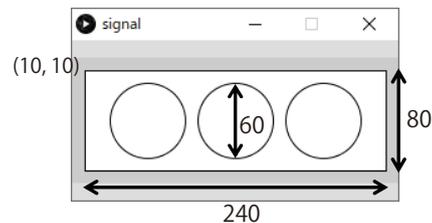


quad(x1, y1, x2, y2, x3, y3, x4, y4)

図-11 いろいろな図形を描く関数 (文献1) P.19より)

● 宿題

お手本を参考に、信号機を描いてみましょう。今回はモノクロですが、次号で色を付けていく予定です。



画面のサイズは横260、縦100
円の中心は左から(60, 50)(130, 50)(200, 50)

参考文献

- 1) Reas, C., Fry, B. 著, 船田 巧 訳: Processing をはじめよう 第2版, オライリージャパン(2016).

(2019年10月31日受付)

杉浦 学 (正会員) manabu@kamakura-u.ac.jp

鎌倉女子大学家政学部家政保健学科准教授。慶應義塾大学大学院政策・メディア研究科後期博士課程修了。博士(政策・メディア)。プログラミング教育をはじめとした情報教育に関する研究に取り組む。中高生向けの著書に『Scratchをはじめよう! プログラミング入門 Scratch3.0版』(日経BP社)など。

第12回全国高等学校情報教育研究会 全国大会（和歌山大会）

Next Stage ～次代の担い手を育む情報教育～

肥田真幸

和歌山県教育庁学校教育課 県立学校教育課（兼）義務教育課

全国大会開催への思い

私は、県立高等学校の教諭であった2017年度、本県での全国高等学校情報教育研究会（以下、全高情研）全国大会の開催を和歌山県情報教育研究会（以下、県情研）に提案した。同じ思いの教員の賛同を得て、大会開催に至った。

本県の情報教育の気運を高めるため、全高情研全国大会を和歌山の地で開催することは、私にとって以前からの目標の1つであった。そのため、2018年度に教育委員会へ異動した後も、実行委員の一人として情報教育の推進に邁進し、第12回全高情研全国大会（和歌山大会）の運営にかかわってきた。

本大会は、大会運営をはじめ、たくさんの方々の協力のもと、県内外から多くの方々に参加をいただき、盛大に開催することができた。

本稿では、本大会について報告するとともに、全国大会を地方で開催する意義や今後の展望、本県の情報教育の取組について述べる。

第12回全高情研全国大会（和歌山大会）の報告

平成から令和に改元し、新しい時代が幕開けた今年（2019年）、8月10日・11日の2日間の日程で開催された。

主催である全高情研は、高等学校で「情報」を教える教員の情報交換を目的とした団体である。全国の情報科担当教員や情報教育の関係者が事例発

表や情報交換を行う場として、年に一度、全国大会を開催している。

本大会は、第12回大会であり、国立大学法人和歌山大学を会場として開催し、全国から378名が来場した。この参加者数は、東京都で開かれた第10回の記念大会に次ぐ人数であり、地方開催では最多となった。

開会行事では、県情研会長・大会実行委員長である県立和歌山工業高等学校長 西村文宏氏による開会宣言の後、全高情研会長・東京都立田無高等学校長 山下一郎氏から開会の挨拶があった。続いて、来賓として和歌山県教育委員会教育長 宮崎泉氏、国立大学法人和歌山大学長 伊東千尋氏から祝辞をいただき、大会が幕開けした（図-1）。

今大会のテーマは、「Next Stage ～次代の担い手を育む情報教育～」である。これは、第1回大会のテーマである「Next Stage —新たに広がるネッ



図-1 開会式の様子



トワークの構築—」を参考にして掲げたものである。情報教育が次のステージに進み、本大会が次代の「はじまり」として、新しいスタートを切るという思いが込められている。

本テーマにあるように、情報科にとって新しい時代の「はじまり」ともいえる新学習指導要領の実施を見据えた内容を含んだ実践事例等が、28の分科会(図-2)ならびに24のポスターセッションで発表された。

本大会の分科会、ポスターセッション発表では、特に「プログラミング教育」に関する事例が多く、注目を集めた。新学習指導要領ではプログラミング教育が小学校から必修化され、中学校でプログラミング教育に関する内容が充実し、高等学校「情報Ⅰ」でも必修化されるといった動向の中、全国の情報科担当教員のプログラミング教育への注目度はより高まっていると考える。今回、本県からも、本県が推進するプログラミング教育について分科会発表を行った。

また、「情報Ⅰ」の年間指導計画や、新学習指導要領の内容として充実が図られているデータ利活用、情報セキュリティ、情報モラル、情報デザイン等に関する事例発表にも注目が集まった。

新学習指導要領における情報科目の授業の検討、実施に向けた準備が各地域で始まり、次代を担う生徒に必要とされる資質・能力の育成のためには、これまで以上に教員の高い専門性が求められている。



図-2 分科会発表の様子

各分科会において発表後の質疑応答や情報交換も活発に行われ、参加者にとって新しい課題の発見や解決につながる有意義な時間となった。

基調講演では、東京大学・慶應義塾大学教授(元文部科学大臣補佐官)鈴木寛氏による、「AI時代の教育」と題したご講演をいただいた。その中で、国内外の教育に関する調査報告や動向、AI時代に求められる資質・能力、論理的思考力や情報活用能力の重要性、また情報産業への人材育成が急務であること等、Society 5.0に向けた教育や人材の育成等の課題と展望についてお話をいただいた。

基調講演に引き続き、文部科学省国立教育政策研究所教育課程研究センター研究開発部教育課程調査官 鹿野利春氏による講評講演が行われた。講演では、社会の変化に伴う情報教育の進展について触れられ、情報科担当教員として、不断の研究や教育実践、その蓄積、そして共有が大切であるといった内容のお話をいただいた。

閉会行事では、大会実行委員長の挨拶の後、次の開催県である愛知県情研会長・愛知県立安城東高等学校長 花井和志氏による挨拶があった。続いて、神奈川県高等学校教科研究会情報部会長・神奈川県立横浜立野高等学校長 菊地勇人氏が閉会のことばを述べ、大会は幕を閉じた。

全国大会を地方で開催する意義

本大会を本県で開催したことによって、地方での全国大会開催はとても意義があると感じた。それは、全国大会開催が、本県の情報科が抱える課題解消に向けた、効果的な一歩となったからである。

情報科については、本県を含み次のような課題を持つ地域が多い。

- 情報科を専門とする教員が少ない。
- 次年度に教科担当者が代わる学校もあり、教科研究部会や研修会等への継続的な参加者が少ない。
- 各学校における教科担当者が一人であることが多

く、授業の内容について、校内で相談しにくい。

各地域が抱えるこれらの共通した課題は、情報教育を推進しようにも、各学校の担当教員の意識を高めにくいといった状況を生む。全国大会の開催によって、これらの課題の解決に向けた、いくつかの糸口を見出すことができた。

その1つ目は、これからの教育において、情報科の授業の必要性がより高まっていることを参加した教員が感じてくれたことである。本県では情報科専門教員が少ない中、各学校から情報科担当教員等が大会運営スタッフとして、50名近く参加してくれた。他県スタッフからも、「県内スタッフのこの人数は前例がない。和歌山県の運営は力強い」と称賛の声をいただいたほどである。県内スタッフとして参加した教員が、運営にかかわりながら、全国の授業実践や動向を知ること、「情報Ⅰ」「情報Ⅱ」という新科目への期待とそれを効果的に実施していこうとする意識がより高まったように感じる。

また、本県では今年度(2019年度)から、「きのくにICT教育」(※内容は最後に触れる)の推進を本格的に開始しており、各学校でプログラミング教育をはじめとする情報教育を推進している。こうした中、今回の全国大会の開催は、プログラミング教育を担当する教員の意識をより前向きに変えるといった相乗的な効果をもたらした。

2つ目に、教員のコミュニティが活性化されたことである。前述したとおり、本大会の参加者数は、378名であった。これは、実行委員として尽力してくれた教員が目標を1つにし、大会の開催を広く周知した成果である。

また、実行委員会の母体である県情研に所属する教員が、前年度から会議を重ねるうちに、教員間のコミュニケーションがより活性化され、つながりが強くなった(図-3)。こうした中、各教員の大会の成功を切望する思いが高まり、さまざまな場面で、積極的に情報を共有していこうという動きが生まれた。このことが何よりも有益であった。

さらに、全高情研のメンバから大会運営のノウハウを得る中で、本県教員と他地域の教員とのつながりが広がった。これは、今後、県内の情報教育を進めていく上で、強みとなるであろう。

3つ目に、情報科の授業の実施にあたって、教員が主体的に取り組む意識をより高めたことである。実行委員会のメンバをはじめ、スタッフとして参加した県内の教員が各分科会やポスターセッション、基調講演や講評講演を通じて、情報教育の全国的(世界的)な動向を「自分事」として捉えてくれた。このことによって、多くの情報科担当教員が、AIリテラシー教育やデータ利活用、プログラミング教育等の必要性やこれからの社会におけるその役割を主体的に考え、子供たちに教えるべき内容を更新していく必要があることを再認識することができた。

本大会は、本県の高等学校における情報教育推進のまさに起爆剤となり、情報教育を重視し進めていくきっかけとしても重要な意味を持つものになった。私は、このきっかけを土台として、これからも県内の情報科担当教員の意識・専門性・教科指導力の向上ために邁進し、本県の情報教育を牽引していきたい。

地域の教科研究部会の意義

地方における研究部会が果たす役割は大きいものがある。県内では、県情研が教科研究部会の役割を担い、情報科の授業の質向上を図ってきた。



図-3 全国大会本部での打合せの様子



県情研の全高情研への加盟は、まだ歴史が浅いが、全国大会の開催を成功に導く上で、情熱を持った教員とそれを束ねる県情研の存在および功績は大きかった。他地域においても、教科研究部会をはじめとした情報を共有できる機会やコミュニティの構築を図ることは、情報教育の推進にとって有効な手立てになると考える。情報科担当教員の指導力向上や教員同士の情報共有、相談しやすい環境づくりは、情報教育が要となるであろう次代の教育の充実に向けて非常に重要な要素であり、次代を担う子供たちの育成に大きな影響を与えるものである。

和歌山県「きのくに ICT 教育」

最後に、本大会でも発表させていただいた、本県が推進している「きのくに ICT 教育」を紹介したい。本県では、新学習指導要領の実施に先立ち、今年度(2019年度)から県内すべての小・中・高等学校および特別支援学校で、発達の段階に応じた体系的なプログラミング教育を実施している。

小学校では、5・6年生で各8時間、中学校では技



図-4 きのくに ICT 教育・学習指導案

術・家庭科(技術分野)で3年間を通して合計25時間、高等学校では共通教科情報科の授業で20時間程度のプログラミング教育を、小・中・高等学校で体系的に実施することとしている。

また、小・中・高等学校それぞれのプログラミング教育に関する学習指導案を県教育委員会で作成し、プログラミング教材とともに配布した(図-4)。さらに、校種ごとの教員研修の実施やプログラミング教育支援員等の学校への派遣、公開授業等の実施など、プログラミング教育の推進に努めている。

「きのくに ICT 教育」を通して、次代で活躍できる情報活用能力に長けた人材が本県から多く育つことをめざしている。

きのくに ICT 教育に関する情報

● 学習指導案

<https://www.pref.wakayama.lg.jp/prefg/501100/ictforum.html>

● 和歌山県教育広報テレビ番組「はばたく紀の国～教育は今～」

<https://www.pref.wakayama.lg.jp/prefg/500100/kouhou/habatakukinokuni/habatakukinokuni.html>

<https://www.youtube.com/watch?v=sTIX0ZJ9mko&feature=youtu.be>

(2019年11月8日受付)

肥田真幸 hida_m0001@pref.wakayama.lg.jp

2012年、高等学校教員として和歌山県に採用される。現在、和歌山県教育庁学校教育局県立学校教育課(兼)義務教育課に所属。情報科担当指導主事として情報教育、ICT教育の推進等を担当している。