

# 共通教育におけるプログラミング未経験者向け入門講義

石黒 克也, 佐々木 正人, 遠藤 隆俊

高知大学 学術情報基盤図書館

ishiguro@kochi-u.ac.jp

## Introductory lectures for inexperienced programmers in common education

Katsuya Ishiguro, Masato Sasaki, Takatoshi Endo

Library and Information Technology, Kochi Univ.

### 概要

高知大学では、2016 年度よりプログラミング未経験者向けの入門講義「初等プログラミング入門」を共通教育において開講している。主な目的はプログラミングの基本的な考え方を身につけること、およびプログラミングに対する潜在的な苦手意識を弱め、興味を持たせることである。また一昨年度より開始されている、小中高におけるプログラミング学習を取り入れた新学習指導要領に基づく教育について知ってもらうことも目的の一つである。本稿ではこれまでに「初等プログラミング入門」で実践してきた内容を報告し、今後の共通教育におけるプログラミング教育を検討する。

### 1 はじめに

「Society5.0」、「第4次産業革命」などをキーワードとして、2016年、初等中等教育におけるプログラミング教育の必修化が決定された [1]。2020年の小学校 [2] を皮切りに、プログラミング教育を取り入れた新学習指導要領の実施が順次開始され、中学校は2021年、高等学校では2022年から実施されている。また、2017年には初等中等教育におけるプログラミング教育等のIT・データ教育の実装 [3] も閣議決定されている。近年では“デジタル社会の「読み・書き・そろばん」である「数理・データサイエンス・AI」の基礎などの必要な力を全ての国民が育み、あらゆる分野で人材が活躍“することを目標として、大学・高専での「初級レベルの数理・データサイエンス・AIの習得」を必修化する取り組みが開始されている [4]。2022年になってからも、政府はデジタルの力を活用した地方の社会課題解決のため「デジタル田園都市国家構想基本方針」を定めており [5]、構想実現に向けた取り組みに「デジタル人材の育成・確保」を挙げ、その中の一つに「高等教育機関におけるデジタル人材の育成」を謳っている。

高知大学（以下、本学と記す。）も多分に漏れずこれらの取り組みに大きな影響を受けており、データサイエンスやAIに関する新たな授業が開始されている。本学では、全1年生に対して第1学期に共通教育で情

報処理の授業を必修化しており、そこでは主に大学のPCの利用方法（ネットワーク接続、メールなど）、情報セキュリティ、情報倫理、ワード、エクセル、パワーポイントなどの一般的な使い方が教えられている。今後「初級レベルの数理・データサイエンス・AIを習得」することが必修化されることを考慮すると、この情報処理の枠組みの中で、あるいはそれを拡張することにより、プログラミングやAIなどの事柄に触れられる機会を作るのが必修化に対応するひとつの方法だと考えられる。授業時間や教員の確保等の問題のため、すぐには対応できないのが現状であるが、2020年度より情報処理の2回分をAI・データサイエンス教育に充てて実施している。

プログラミング教育については、2016年度より共通教育の教養科目として「初等プログラミング入門」が開講されている。前述の情報処理の授業アンケート結果によると、学生の中には初歩的なプログラミングを学びたいという潜在的な需要が存在しており、それらを踏まえ、「初等プログラミング入門」は共通教育情報処理に続くプログラミング未経験者向けの授業と位置付け開講されている。特に本授業は共通教育の教養科目に分類されており、プログラミングを専門的に学ぶ準備というよりは、プログラミングとはどのようなものであるのかを広く浅く学ぶことを主眼としている。

2016年度および2017年度の授業では、プログラミ

ング言語として入門用言語である「Sunaba」を用いた [6]。その後、初等中等教育においてプログラミング教育が必修化されることが決まったことを考慮して、2018 年度からはプログラミング環境を Sunaba から「Scratch」 [7] に変更して実施している。これは、Scratch がプログラミングの考え方を学ぶのに優れた教材であること、またプログラミング教育必修化後の教育を受ける今後の学生の多くが学ぶことになるであろう Scratch を、必修化教育を受けていない現在の学生が学ぶ機会を設けることも必要であろうと考えてのことである。

本稿では、2016 年度から実施している「初等プログラミング入門」の経験 [8] を元にして、これまでの未経験者向けプログラミング入門講義についての実践報告、および今後のプログラミング教育についての検討を行う。

## 2 初等プログラミング入門

### 2.1 Sunaba

「初等プログラミング入門」の授業では、2016 年度からの 2 年間はプログラミング言語として平山尚氏により作られた「Sunaba」を用いた [6]。この言語を選択した理由は次のようなものである。

- エラー表示を含め、日本語が使えるので初心者にわかりやすい。
- 基本的な機能のみしか持たず、プログラムを基礎的な部分から構成することになり教育的である。
- コンパイルの作業が必要なく、ドラッグアンドドロップで即時実行できる。

授業では簡易テトリスの作成を最終目標とした。簡易テトリスは、枠の中でテトリミノ（4 つの正方形が繋がったもの）が上から落ちてきて、そのテトリミノをキーボードを使って左右移動や回転させることができるものである。授業の流れと内容を次に示す。

1. Sunaba インストール、動作確認
2. 点を打つ、四角形を書く
3. 繰り返し構文
4. 部分プログラム、枠の作成
5. 部分プログラム
6. 四角形を落下させる
7. キーボードを使って点を動かす
8. 条件実行文
9. 四角形を積もらせる

10. 一列に並んだ列を消す
11. 落下する四角形を 2 つにする
12. 2 つの四角形を回転させる
13. 落下する四角形を 3 つに拡張する
14. 落下する四角形を 4 つに拡張する
15. まとめ

授業時には本学独自の教務情報システム (KULAS) の機能を活用し、資料配布、課題提出、出席確認、アンケートなどを実施した。毎回講義用スライドを授業数日前に KULAS に載せることとし、スライドには前回の復習と次に学ぶ新しい内容を含めた。これは授業の課題をなるべく授業中に終わらせられるよう事前学習（復習と予習）を促すために行ったもので、反転学習となることを意識している。

Sunaba はプログラミングに慣れているものには日本語と半角でのタイピングがわずらしく、限られた機能しかないため使いやすいとは言い難いが、初心者にはわかりやすい言語であった。また実行時のエラーがわかりやすいため、プログラミング未経験者でも自分で問題を解決しやすいようであった。一方教える側からすると、機能が限られている分同じことをするにも工程数が多くなるのでまどろっこしく感じられるが、逆に考えると順にステップを踏むことになるため、ある意味教えやすく感じられた。Sunaba を使って一つのゲームを作るという作業を通して、プログラミングの考え方の一端には触れられるものとなったのではないかと思われる。

### 2.2 Scratch

2016 年度および 2017 年度の授業では、プログラミング言語として入門用言語である Sunaba を用いた [6]。プログラミングの考え方を学ぶには Sunaba でも特に問題はないが、入門用に特化した言語であるため、プログラミングの世界がその中で閉じてしまいがちである。プログラミングを学ぶためには他のシステムとの連携などを考えることも有効であり、また本授業は教員志望の学生も受講することから、初等中等教育の授業等で使うことのできる知識や技術を身に付けることも大事になってくる。それらを考慮して、2018 年度からは教材として「Scratch」 [7] を使用している。Scratch を選択した理由は、文法を知らなくても直感的に使うことのできるビジュアルプログラミング言語であること、Web アプリケーションとなっているため特別な環境構築が必要なく、使用者の OS 等の環境に依存せず利用可能なこと、コンパイル作業が

必要なくすぐに動作確認できること、作成したプログラムと実際の動作の対応がわかりやすく、プログラムのミスを見つけやすいこと、などである。受講生が目指す授業目標には、1. プログラミングの概念を理解する、2. 基本のアルゴリズムを理解する、3. Scratch でプログラミングできるようになるの3つを掲げた。

2018年度および2019年度の最初の2年間は、試行錯誤しながら授業の内容および流れを作っていた。大まかな内容は次のようなものである。まず、Scratchの画面構成や使い方がある程度説明し、その後はScratchの基本的な機能を使って繰り返し構文やif構文、乱数の使い方などを学習する。具体的にはメッセージ送受信機能を使って登場人物を時系列に沿って動かす、変数や配列の使い方を学ぶ、探索アルゴリズムや並べ替えアルゴリズムなどを理解する、定義機能(関数機能)を学び応用する、といった内容である。授業資料はLMSに載せることで、自分のペースで理解を深めてもらうようにした。また毎回サンプルファイルとして課題の元となるScratchファイルを提供し、それを使用することによって細かいことに囚われず、各回で新しく学ぶ内容に集中できるように工夫した。これにより、授業中にまったく手が動かないといった学生は皆無となり、多くの学生は授業中に課題を終えることができるようになった。

2019年度にはScratchを学んだことの総仕上げとして、ドローンプログラミングも実施した。ドローンプログラミングにはRyze Technology社のトイドローン「Tello Edu」を用いた。プログラミング入門の授業においてドローンプログラミングを導入した理由は、プログラミング未経験者にとって、プログラミングにより何かを物理的に動作させる経験は、プログラミングに対するイメージをより明確にすることに役立つであろうと考えてのことである。また、「Tello Edu」がScratch(およびPython、Swift)によるプログラミングを公式にサポートしていることも大きな理由である。

2020年度からは全世界的なコロナ禍の影響により、大学の授業も対応を余儀なくされ、それは現在にも影響を及ぼしている。特に2020年度はほとんどの大学において、授業はオンライン対応を迫られ、初等プログラミング入門の講義もそれに従って実施された。オンライン講義に利用したシステムは、moodle(LMS)、KULAS(教務情報システム)、Webex(オンライン会議システム)などである。毎週の講義ではWebexを使って同期型授業を行い、同時に授業の録画も実施し、

録画された動画をmoodleにアップロードすることによって非同期対応も可能なようにした。また、資料(PDF資料、プログラム動作イメージ動画など)を事前にmoodleに載せることにより、反転学習も可能な状況とした。全15回の授業内容は以下の通りである。

1. オンライン授業の説明、Scratchの使い方
2. 画面構成、座標
3. 乱数
4. メッセージ送受信
5. 変数
6. リスト(配列)
7. アルゴリズム 1
8. 場合分け
9. ゲーム作成 1
10. 定義、引数
11. アルゴリズム 2
12. 乱数、グラフ化
13. モンテカルロ法
14. ゲーム作成 2
15. 課題

この授業内容は現在もほぼ踏襲されており、これまでの内容に新たに乱数を使ったデータ生成およびそのデータのグラフ化、モンテカルロ法などが加わっている。データ生成やグラフ化には数値計算の基礎的な考え方を知る必要があるが、それまでの内容ではこれらのことにあまり触れていないこともあり、動作させるために苦労した学生も多くいたようであった。

2019年度から導入したScratchと外部機器を接続してプログラミングを学ぶ内容については、PCと機器を実際に接続して操作する必要があるが、コロナ対応によりオンライン授業を行わざるを得なかったため、2020年度以降の授業ではあまり行うことができていない。しかしながら、大学から対面授業実施の許可が出たタイミングを見計らい、受講生から希望者を募って数回実施している。希望者のみとしたのは、オンライン授業のため実家に帰っており大学に出てこれない学生がいること、許可が出たといってもコロナ禍において対面授業を望まない学生がいること、などを考慮してのことである。希望者のみとしたことにより、参加の有無は成績には含まないこととした。用いた外部機器はトイドローンとmicro:bitの2つである。トイドローンについては前述したので、ここでは「micro:bit」について概要を述べる。micro:bitはイギリスのBBC(英国放送協会)が主体となって作った、

手のひらサイズの教育向けマイコンボードのことである。その特徴としては

- LED や無線アンテナ、センサー（光・温度・地磁気・加速度）などを搭載
- パソコンやタブレットなど、様々な環境でプログラミング可能
- 様々な拡張パーツを使用可能

などが挙げられる。本稿執筆時、micro:bit の最新バージョンはバージョン 2.0 であるが、本講義で使ったものはバージョン 1.0 である。micro:bit のプログラミングには Scratch と同じようなビジュアルプログラミング言語である「makecode」を用いた。プログラミングの内容は

- LED を使って特定のマーク（ハートなど）を表示させる
- 乱数や LED、加速度センサーを用いてじゃんけんゲームを作成する
- 光センサーや温度センサーを用いて、その場の明るさや温度を画面に表示する

などの基本機能を用いた簡単なものである。makecode はブロックを組み合わせてプログラミングを行うが、Scratch をある程度使えるようになっていれば、特に説明などがなくても感覚的に使用できる。そのため、Scratch の授業がある程度進んでから micro:bit の用いれればスムーズに導入可能であった。あまりに簡単に使えたため、参加した学生も驚いたようであるが、PC と外部機器をつなぎ、外部機器が動作している様子を見ることにより、プログラミングが PC の画面の中だけでなく外の世界とつながれることを実感することができたとの感想もあり、当初の狙いはある程度達成されたものと思われる。

また、近年の AI・データサイエンス教育の高まりを意識して、2021 年度から Scratch を使って機械学習を体験する内容を最終課題として授業に取り入れた。Scratch と機械学習の連携に用いたのは「ML2Scratch」[9]である。「ML2Scratch」は Google の開発した機械学習用ソフトウェアライブラリ TensorFlow と Scratch をつなぐ仕組みであり、Scratch を使って画像判定などを行うことが可能となるものである。プログラムの構成によっては図 1 のように文字識別なども可能となる。これを用いて、最終課題では手書き図形（丸や三角形、四角形など）の識別ができるようになることを課した。最終提出物は Scratch の



図 1 ML2Scratch を用いた手書き数字の識別例

プログラミングファイルと学習済みデータである。この課題はほとんどの学生が問題なく実行できており、データ作成やラベル付けなどを行うことによって、機械学習の一端を学ぶことができたものと思われる。

### 2.3 実践

「初等プログラミング入門」は、2019 年度までは第 1 学期または第 2 学期のいずれかにおいて週 1 回の開講であったが、2020 年度からは第 1 学期および第 2 学期にそれぞれ週 1 回ずつの開講とした。このように変更した理由は、プログラミングを学んでみたいと考える学生はそれなりに存在するものの、年に 1 コマのみの開講では他の必修講義等と重なって受講する機会を失ってしまう学生がいることを考慮してのことである。授業のコマ数を増やしたものの、基本的に対面式の授業を検討していたこと、TA/SA が付かないこと、採点の手間などのことを考えて、目配りが可能と思われる程度に人数制限（50 人）を課すことにした。履修登録時には毎回抽選が行われるほど受講希望者がおり、年に 100 人ほどが受講していることになる。

講義ではまず受講生の状況を把握するため、第 1 回目の講義において、利用している OS やプログラミング経験などについての事前調査を行っている。初等プログラミング入門の授業を開始した 2016 年度、あるいは Scratch を使い始めた 2018 年度からの調査結果の主なものを以下に示す。

1. 受講生のうち、これまでに何らかのプログラミング経験があるかどうかを尋ねた結果が図 2 である。初等プログラミング入門の授業当初から、全体の傾向はあまり変わらず、プログラミング経験があると答えた学生の割合は 30 % 以下に留まっている。プログラミング経験の内容では、中学や高校で Scratch を使った、ロボットプログラミングをしたなどの学生が多かった。
2. Scratch について知っていましたか？という問いに対する回答が図 3 である。2018 年には知っていた学生の割合が 10 % 程度で、この傾向は 2020

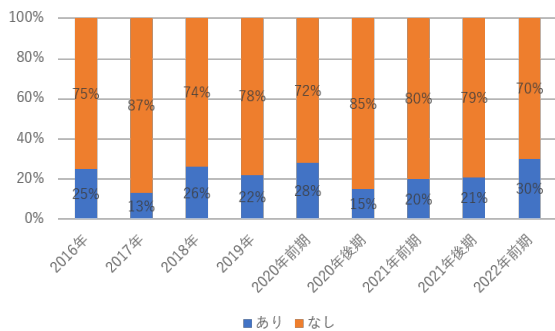


図2 これまでにプログラミングの経験は？

年まではほとんど変わっていないが、2021年度あたりから20%を超え、2022年度前期には30%以上にまで上昇している。このことは、2020年度から開始されたプログラミング教育必修化の影響により、Scratchが多くの人に認知され始めたことが原因ではないかと推測している。

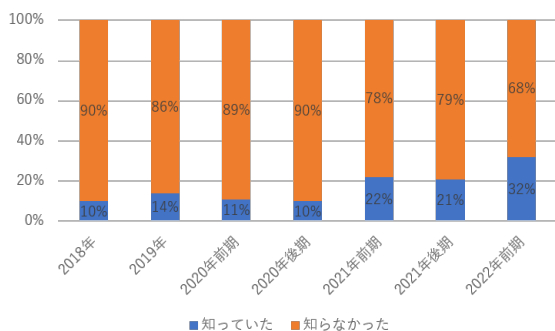


図3 Scratchのことを知っていましたか？

## 2.4 授業後アンケート

本講義の終わりには、受講生に対して授業内容やScratch、オンライン授業などについてのアンケートを実施した。以下、授業後アンケート結果の一部を紹介する。

- Scratchは初等プログラミング入門の授業で用いるのに適当だと思いますか？との問いには、全体の90%から95%の学生が適当またはそこそこの適当と回答している(図4)。その理由としては、命令ブロックが色分けされていてわかりやすく、また画面表示が日本語対応しているため、プログラミングが簡単を感じる、ブロックの組み合わせ次第で複雑な動きを実現でき、ゲームなどの作成もできるため、最後まで脱落することなく続けられた、などが挙げられている。
- 本講義ではmoodleを使った資料提示、同期型講

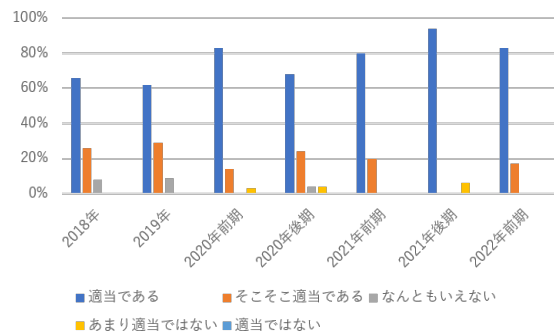


図4 Scratchは初等プログラミング入門の授業で用いるのに適当だと思いますか？

義および非同期型講義を実施しましたが、プログラミングを学ぶ授業としてこの方式はどうだったでしょうか？との問いには、図5のような回答であった。コロナ禍当初は同期型授業を希望する学生もある程度いたが、コロナ禍が長引き、オンライン授業に慣れてくるにつれ同期型授業のみでよいという学生は減少しているようである。学生のスキルにも依存すると思われるが、資料のみで十分という学生や、資料だけではイメージがわからない場合には動画を見返すという学生も存在する。より多くの学生に対応するには、同期型も非同期型もともに準備することが望まれていると考えられる。

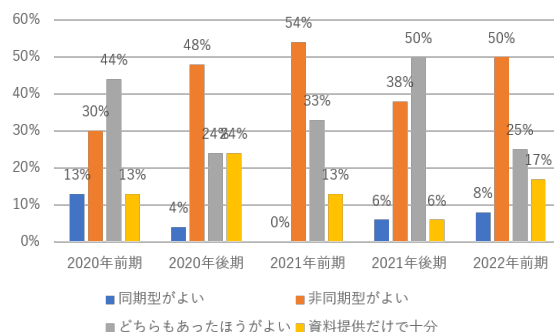


図5 本授業ではmoodleを使った資料提示、同期型講義および非同期型動画講義を実施しましたが、プログラミングを学ぶ授業としてこの方式はどうだったでしょうか？

- もし初等プログラミング入門の次のステップのプログラミングの授業があれば、受講してみたいと思いますか？との問いには、80%から90%が「思う」あるいは「わからない」と回答しており、Scratchの次のステップとなる内容の授業についても検討していく必要があるようである。(図6)。



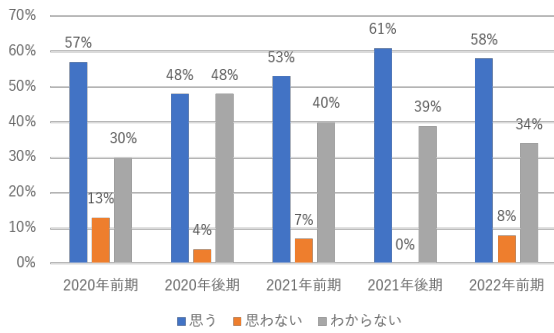


図6 もし初等プログラミング入門の次のステップのプログラミングの授業があれば、受講してみたいと思いますか？

以上のアンケート結果を見る限りでは、本授業の内容に関しては初等プログラミング入門の授業目的に合ったものとなっており、またプログラミング初心者からの脱却を目指す次のステップの授業を検討することも重要であろうと思われる。

### 3 おわりに

本稿では、2016年度から実施しているプログラミング未経験者向けの授業「初等プログラミング入門」の実践報告を行った。本講義では2016年度からの2年間は「Sunaba」を、2018年度からは「Scratch」をプログラミング言語として用いている。開始当初は単に未経験者向けの授業を行うことを考えていたが、同時期に初等中等教育におけるプログラミング教育必修化が決まり、徐々にそれを考慮した授業にシフトしていった。これまでは、一部の職業に就くものを除けば、プログラミング未経験者ということでも特に問題はなかったのであるが、今後は新しい教育を受けた世代がプログラミング経験者となり、基礎的なプログラミングの知識を持っていて当然のような時代になることが予想される。このような状況を踏まえると、新指導要領に沿って学んだ学生とそうではない学生の間には、プログラミングに関する知識の差が生じることが容易に想像される。特に新指導要領のもとで学んだ最初の学生が大学に入学するのは2025年度からであり、そこで大きな差が生じるのではと予想されている。そのため、その差をなるべく埋められるよう「初等プログラミング入門」の内容を吟味し、変化させてきた。さらに大学において数理・AI・データサイエンス教育の必修化が進められることも決まり、それらを意識したプログラミング入門講義へと内容の充実化を図っている。

しかしながら、前述したように2025年度には新しい指導要領に基づいた教育を受けた学生が入学してくることになり、プログラミング未経験者は（理想的には）皆無に近い状況になることが予想される。そうなると、これまで行ってきた講義の前提条件は大きく変化し、プログラミング入門の内容も経験者向けのものとして、見直さざるを得なくなってくると思われる。つまり、現在の授業内容は続けてもあと2年ほどであり、3年後にはこれまでとは異なる内容での授業となるよう検討していかなければならない。今後は学内のデータサイエンス教育の進展を注視しつつ、2025年度入学生の状況把握に努め、それらに合わせた新しい授業を模索する必要があると考えている。

### 参考文献

- [1] 日本再興戦略 2016, [http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/2016\\_zentaihombun.pdf](http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/2016_zentaihombun.pdf), (2022年10月10日閲覧)
- [2] 文部科学省, [https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1375607.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1375607.htm), (2022年10月10日閲覧)
- [3] 未来投資戦略 2017, <http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2017.pdf>, (2022年10月10日閲覧)
- [4] 総合科学技術・イノベーション会議(第43回), <https://www8.cao.go.jp/cstp/siryo/haihu043/haihu-043.html>, (2022年10月10日閲覧)
- [5] デジタル田園都市国家構想基本方針, [https://www.cas.go.jp/jp/seisaku/digital\\_denen/pdf/20220607\\_honbun.pdf](https://www.cas.go.jp/jp/seisaku/digital_denen/pdf/20220607_honbun.pdf), (2022年10月10日閲覧)
- [6] 石黒克也、佐々木正人、佐々浩司、“共通教育におけるプログラミング入門講義”、2017年度大学ICT推進協議会年次大会論文集、WA2-1(2018)
- [7] Scratch, <https://scratch.mit.edu/> (2022年10月10日閲覧)
- [8] 石黒克也、佐々木正人、佐々浩司、“共通教育における教養科目としてのプログラミング入門”、2019年度大学ICT推進協議会年次大会論文集、TH1-5(2020)
- [9] ML2Scratch, <https://github.com/champierre/ml2scratch> (2022年10月10日閲覧)