

統計学を題材にしたプログラミング入門講義

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 programming lecture on statistics

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 2020 年度より順次開始されている、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簡単に振り返り、今年度から始めた統計学を題材にした新しい内容での「初等プログラミング入門」について報告する。

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 年に政府は初等中等教育におけるプログラミング教育の必修化を決定した [1]。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 [2] も閣議決定されている。これらを受けて、各種指導要領にはプログラミング教育等が記載されることとなり、2020 年度より小学校、中学校、高等学校で順次プログラミング教育が開始されている。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 [3]。これに伴い、文科省は「数理・データサイエンス・AI」認定制度（リテラシーレベル・応用基礎レベル） [4] を作り、これらの基礎的な能力の向上を図る機会の実質的な拡大を支援している。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。各大学においては、これらの認定を受けるための人員調整を

含めた体制作り、シラバス作成などが行われ、準備が整ったところから順次新しい授業を開始している。

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、そこでは主に大学での 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おり、2024 年度からは授業内容を大幅に刷新して対応していく予定である。

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] に変更して実施した。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重要であろうと考えてのことである。2023年度からは内容を刷新し、プログラミング言語として「Python」を用いることにした。新しい学習指導要領で学んだ学生が2025年度に大学に入学してくることを踏まえると、これまでのようなプログラミング完全未経験者向けの授業だけでは学生の要望に応えられない可能性も考えられるため、より幅広い内容を初心者でも比較的取り組みやすい言語で実施できるように検討してのことである。また、単にプログラミング言語を変更したのみならず、授業の題材として「統計学」をメインに据えた。前述したように、大学においては「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行われており、その中では統計学の理解が重要になるためである。

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、本年度に開始した新しい内容でのプログラミング入門講義についての実践報告を行う。

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。

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] を使用した。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。

授業の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。

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

2019年度からはScratchと外部機器を接続してプログラミングを学ぶ機会を設けた。用いた外部機器はトイドローンとmicro:bitの2つである。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することが可能となるものである。

2.3 Python

「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。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1である。初等プログラミング入門の授業開始当初の2016年度から、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。



図1 これまでにプログラミングの経験は？

2. 「Python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2である。プログラミング未経験者がほとんどの授業なので約4割しか知らなかったという結果になっているが、2022年度まで行っていたScratchでは多いときでも知っていると答えたものが3割程度であったので、個人的

には認知度は高めという印象を持った。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、Pythonが多くの人に認知され始めたこと、あるいはAI・データサイエンスについて調べると必ずといってよいほど出てくるキーワードになっていることが原因ではないかと推測している。

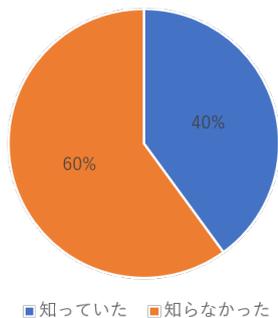


図2 Scratchのことを知っていましたか？

3. 「小中高でプログラミング教育が必修化されていることを知っていましたか？」という問いに対する回答が図3である。いずれの場合も、5割から6割が知っていたと回答しており、受講生の多くは今後の大学や社会において、プログラミングの知識やスキルが重要になってくるものと認識しているようである。

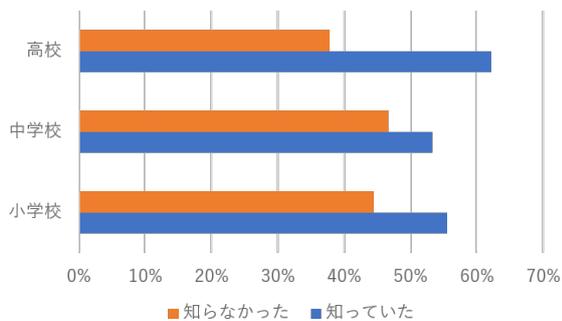


図3 小中高でプログラミング教育が必修化されていることを知っていましたか？

2.4 授業内容

本講義ではプログラミング言語として「Python」を用いた。プログラミング環境には「Google Colaboratory」を採用した。Pythonを動作させる環境には様々なものがあり、BYODを実施している本学では各自のPCにPythonをインストールして利用することも考えられたが、PCの仕組みを知らない初心者にとって環境構築には手間がかかることが多い。この手間が

原因でプログラミングそのものへの興味を失ってしまうことは避けたいと考え、誰でも手軽にPythonを利用できるGoogle Colaboratoryを用いることとした。また、多くの他の大学ではGoogle Colaboratoryの環境ではパワー不足で十分な作業が行えない状況である、ということも聞いていたが、動作させるプログラムがいずれも簡単なものであり、実行時間も数秒で済むものがほとんどのため、専用のサーバを構築するという選択は行わなかった。Google Colaboratoryを利用するためのアカウントは、各自で取得し、各自で管理するようにした。アカウント作成時には様々な情報を入力する必要があるが、その内容についても各自に任せることにした。すでにアカウントを持っている場合にはそれを使ってもよいし、授業用に新たに作成してもよいとし、必要以上の内容は入力する際に注意するように促した。

講義内容は「統計学」を主な題材とした。統計学は文系・理系を問わず様々な場面で使われるが、現在はデータをエクセル等に読み込ませてクリックしていくだけで簡単に結果を求めることができるため、その内容を理解して使っている学生はそれほど多くない。また、大学で進められている「初級レベルの数理・データサイエンス・AIの習得」必修化のことを考えると、統計学を理解することは今後非常に重要になってくるはずである。そのため、プログラミングを学びつつ、同時に統計学に親しむことを狙って、「統計学」を題材の軸に据えることとした。特にPythonには統計学のためのライブラリが数多く準備されているため、これらの狙いにマッチした言語と考えられる。

授業の流れは次の通りである。

1. オリエンテーション、プログラミング環境作り
2. Pythonの基礎1
3. Pythonの基礎2（データの読み込みなど）
4. Pythonの基礎3、度数分布
5. 1変量データ（平均、分散、標準偏差、順位に基づく統計量など）
6. 多変量データ（共分散、相関係数、相関行列、様々なグラフなど）
7. 確率分布（ベルヌーイ分布、二項分布など）
8. 確率分布（二項分布、ポアソン分布など）
9. 確率分布（正規分布、中心極限定理など）
10. 統計的推測（データシミュレーション、母集団推定など）
11. 母集団推定（母平均、母分散など）

12. 統計的仮説検定
13. 単回帰分析
14. 重回帰分析
15. ロジスティック回帰

開講当初は上記の他に、将来の応用を見据えて機械学習の初歩的な内容も行うつもりであったが、想定していたよりも内容が多くなってしまい、そこまでは実施することができなかった。また、Python や Google Colaboratory の基礎的な使い方についてはほとんどの学生が特に問題なく使用できたが、1点のみ多くの学生がつまづいた箇所があった。それはデータファイルの読み込みについてである。基本的な流れは

1. データファイルを Google ドライブにアップロードする
2. Google ドライブを Google Colaboratory にマウントする
3. Google Colaboratory からアップロードしたファイルを読み込む

であるが、多くの学生はドライブやディレクトリなどの構造を理解していないため、マニュアル通りに作業を行ってエラーが出てしまうとお手上げ状態になってしまうという状況であった。そのため、ファイルがいまどこにあるのか、どのように探せばよいのかといった基本的なことができるようになるまでに、マニュアルに手を加えつつ 5, 6 回の授業で説明をし、それでもできない場合には個別対応することとした。

授業方法は資料を LMS に載せ、それを参照しながら Google Colaboratory を操作するようにした。まず最初にその日の授業の大まかな流れを説明し、具体的な内容に入っていく。統計学独特の専門用語が出てくるのでその説明をしつつ、数式を使って何を計算しているのかを示し、データを使って具体的な例を見ていく。データは自分で生成する場合もあるが、多くの場合にはこちらで準備したファイルを利用してもらうこととした。データを扱う際には、まずは数式で表された定義通りに様々な量を計算し、その後 Python に準備されている豊富なライブラリを利用して計算する方法を紹介した。ライブラリを使えば結果を得るのは簡単であるが、それでは内容を理解したことにはならないため、定義から出発して理解したうえでライブラリを用いることを推奨することとした。

2.5 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 Python 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。

1. 「これまでに確率や統計学を学んだことがありましたか？」との問いには、全体の 50 % を超える学生が「ない」と回答している (図 4)。特に統計学を学んだことがある学生は 10 % に満たず、ほとんどのものが今回初めて統計学を学んだようであった。また、確率についても学んだことがあるのは 40 % になっている。多くの学生は今回の授業内容のような専門用語が出てくることはなくてもなんらかの形で確率等を学んでいるはずであるが、それと今回の内容が結びついていないようであった。

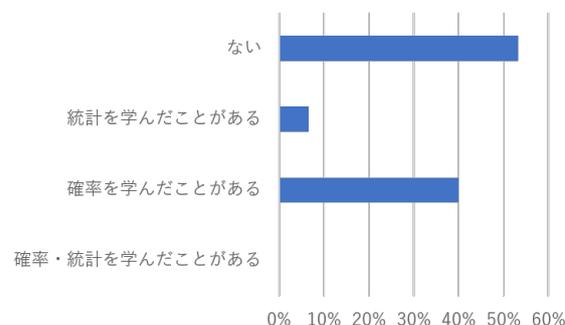


図 4 これまでに確率や統計学を学んだことがありましたか？

2. 「この授業では統計学を学ぶために数学を道具として使いました。数学の次の事柄を学んだことがありましたか？」との問いには、図 5 のような回答であった。数列に関しては 100 % の学生が学んでいたが、対数関数・指数関数や微分・積分は 60 ~ 70 %、行列は約 30 % ほどが学んだことがあるとの回答であった。近年は行列を学ぶ機会がないまま大学を卒業してしまう学生も多いようだが、データ解析等を行う際には必要になってくる場合も多いので、これらの数学的な基礎知識の有無の違いや授業時間内での説明の仕方にも留意することが重要であろう。
3. 「この授業では、今後必須となるデータサイエンス関連科目を考慮して「統計学」を題材として扱いました。「統計学」についてどの程度理解できたと思いますか？」との問いには、図 6 のような回答であった。今後のデータサイエンス必修化を見据

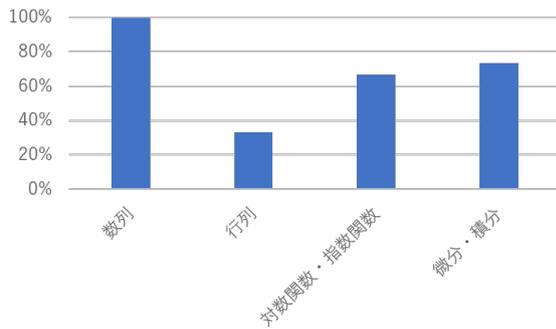


図5 この授業では統計学を学ぶために数学を道具として使いました。数学の次の事柄を学んだことがありますか？（複数回答含む）

えて統計学を題材として用いたが、ある程度まで理解できたと答えた割合は5割強であった。後のアンケート結果にも関連するが、Pythonを使ったプログラミングが難しかったというわけではなく、これまでの設問回答にあったように統計学の数学的な理解が追いつかないことが主な原因だと思われる。これを踏まえると、プログラミング入門の授業で統計学を題材にするのは、学生の知識レベルに依存してしまう部分があり、より多くの学生に平均的な理解を求めるのはなかなか困難な状況であることがわかる。しかしながら、この点については数学的な部分をより手厚くフォローすることにより、ある程度の改善も期待できるものと思われる。

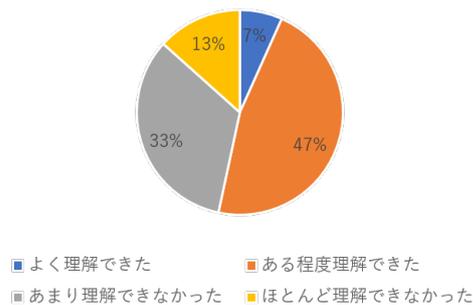


図6 この授業では、今後必須となるデータサイエンス関連科目を考慮して「統計学」を題材として扱いました。「統計学」についてどの程度理解できましたか？

4. 「Python 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、図7のような回答であった。適当でないと答えた学生は10%に満たず、多くの学生はPythonによるプログラミングを肯定的に捉えたようである。

ファイルの読み込みに最終回近くまで苦労した学生もいたが、プログラミング環境として Google Colaboratory を用いたことで、特に大きな問題はなくプログラミングに専念できたようである。ただし、実行時にエラーが出た場合の対処方法がわからなかったという意見が数多く見られたこともあり、今後はこの点についても対応していくことが重要であろうと思われる。

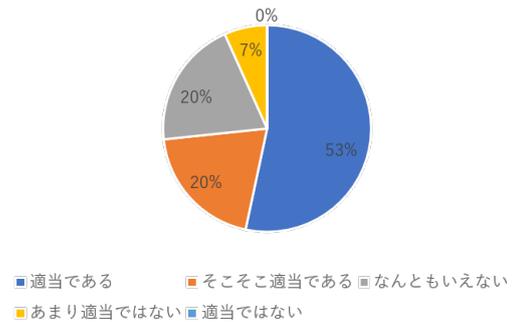


図7 「Python」は初等プログラミング入門の授業で用いるのに適当だと思いますか？

5. 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、70%を超える学生が「思う」あるいは「わからない」と回答しており、より実践的なデータ分析など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りそうである。(図8)。

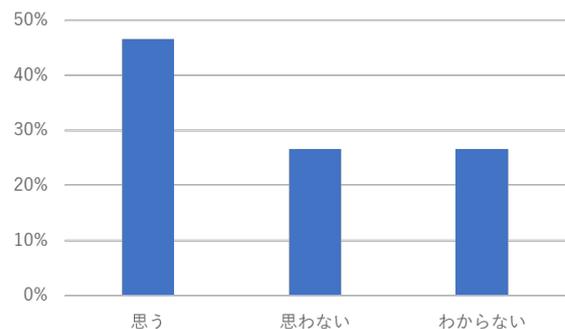


図8 もし初等プログラミング入門の次のステップのプログラミングの授業があれば、受講してみたいと思いますか？

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、そして2023年度からは「Python」をプログラミング言語として用いている。開始当初は単純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。 [10]

参考文献

- [1] 日本再興戦略 2016、http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/2016_zentaihombun.pdf、(2023年9月29日閲覧)
- [2] 未来投資戦略 2017、<http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2017.pdf>、(2023年9月29日閲覧)
- [3] 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/haihui043/haihu-043.html>、(2023年9月29日閲覧)
- [4] 数理・データサイエンス・AI教育認定制度、https://www.mext.go.jp/a_menu/koutou/suuri_datascience_ai/00001.htm、(2023年9月29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、(2023年9月29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/>(2023年9月29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>(2023年9月29日閲覧)
- [10] 石黒克也、佐々木正人、遠藤隆俊、“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”、2022年度大学ICT推進協議会年次大会論文集、13PM2D-4(2023)