研究ノート

情操も育むプログラミング実習

森　　由美

1. はじめに

プログラミング教育は、論理的思考力を育む教育として、これから訪れるSociety 5.0の社会に向けた新しい人材の育成のために注目されるSTEAM教育の１つとして位置づけられており、2020年には小学校での教育が義務化された。これからの小学生が、今後、高等教育を受け、社会人になっていく過程で、プログラミングは読み書きそろばんのように普通の技能の1つとして身につけていくものと考えられる。その一方で、そのような教育を受けてこなかった現・中高生や現・大学生の中には、「プログラミング」と聞くだけで、不安や恐怖まで感じる生徒・学生が少なからず存在する。社会にでるまでに、不安や恐怖などの先入感を払拭し楽しく取り組める体験をしてもらいたい、将来への可能性を広げてほしい、そのような考えで、成城大学・共通科目の「データサイエンス入門Ⅰ」の中でプログラミング実習を実施している。

本稿では、これから訪れるSociety 5.0の社会、この新しい社会を牽引する人材育成として注目されるSTEAM教育、STEAM教育の1つとして重視されるプログラミング教育、プログラミング実習の実践、という流れで、概論・紹介する。

2. Society5.0とSTEAM教育

コンピューターやネットワークによって構築された仮想空間と実世界である現実空間を高度に融合させたシステムによって、経済発展と社会的課題の解決を両立する、人間中心の社会Society 5.0 [1]が訪れようとしている。Society 5.0は、人工知能（AI)、ビッグデータ、IoT (Internet of Things）やロボティクス等の先端技

術が高度化してあらゆる産業や社会生活に取り入れられ、社会の在り方そのものが劇的に変わると言われている。内閣府のホームページでは、Society 5.0 で実現する社会を以下のように図 1 と合わせて説明している [2]。

> “IoT で全ての人とモノがつながり、様々な知識や情報が共有され、今までにない新たな価値を生み出すことで、これらの課題や困難を克服します。また、AI により、必要な情報が必要な時に提供されるようになり、ロボットや自動走行車などの技術で、少子高齢化、地方の過疎化、貧富の格差などの課題が克服されます。社会の変革（イノベーション）を通じて、これまでの閉塞感を打破し、希望の持てる社会、世代を超えて互いに尊重し合あえる社会、一人一人が快適で活躍できる社会となります。”

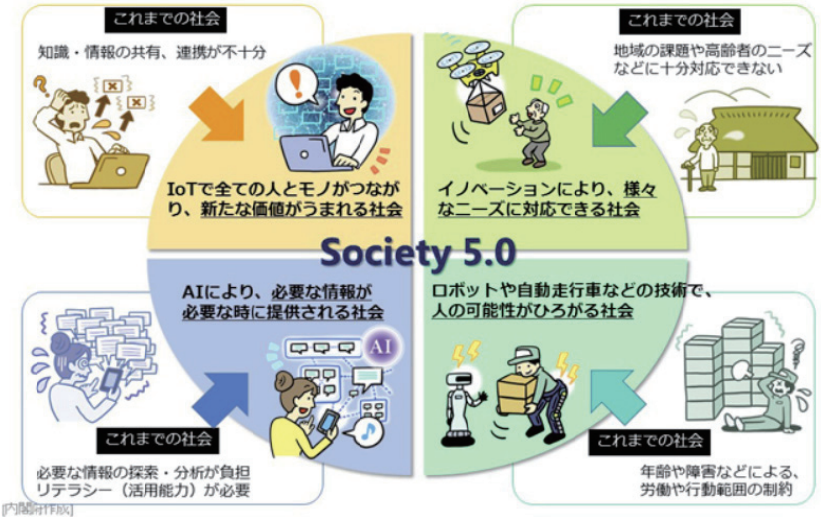

図 1. Society5.0 の社会（内閣府ホームページより）

このような Society 5.0 の社会では、今まで人が行っていた煩雑な作業や簡単にはできなかった作業・危険な作業を、AI やロボットが支援・代行することにより、人々は人でなければできない作業や人の強みを生かした仕事に時間を費やすことができる。逆に言えば、AI やロボットではできない仕事を遂行するための人間

の強みや力を伸ばし発揮することが新たな社会にはさらに求められる。このような新たな社会に向けた人材の育成が急務とされており、このための新たな教育として「STEAM 教育」が世界的に注目されている。

STEAM 教育の前身である STEM 教育はアメリカで生まれ、「Science（科学）、Technology（技術）、Engineering（工学）、Mathematics（数学）の 4 つの分野を統合的に学び、科学技術の分野で活躍できる人材」育成の概念として、特に 2000 年代以降注目され、世界に広まった（各分野の頭文字をとって STEM 教育と呼ばれる）。近年、STEM 教育に Art を含む広い範囲（芸術、文化、生活、経済、法律、政治、倫理等）で「A」を定義し加えた STEAM 教育 [3] が、Society 5.0 に向けた人材育成に必要な教育として推奨されている。2018 年に文部科学省から発表された「Society 5.0 に向けた人材育成 〜社会が変わる、学びが変わる〜」では、大学について次のように人材育成の方向性が示された。

> “ 学生が所属する学部等に関わらず、教育における STEAM やデザイン思考の必要性を踏まえ、学生が必要とする教育をいかに提供していくか、各大学の工夫が期待される ”

Society 5.0 では、「文章や情報を正確に読み解き対話する力」、「科学的に思考・吟味し活用する力」、「価値を見つけ生み出す感性と力、好奇心・探求力」が重要視されており [4]、これらの力も含めて STEAM 教育により新たな社会を牽引する人材を育成していくことが必要とされている。

成城大学のデータサイエンス科目群の 1 つである「データサイエンス入門 I 」では、統計学の入門の講義・データ分析の実習に加えて、STEAM 教育の実践としてプログラミング実習を取り入れている。プログラミング言語として Scratch（スクラッチ）と Python（パイソン）を活用しており、特に Scratch は、これまでプログラミングを体験したことのない学生にとって直感的に扱え、すぐに馴染むことができる言語であり、楽しみながら、プログラミングの基本を学ぶことができる。自分なりの様々な工夫ができることや、作った部分を小分けに逐次確認

して達成感を味わいながら完成に進めていけることなどがこのプログラミング言語の特長であると言える。

3. Scratch プログラミング実習

Scratch は、Scratch 財団が米国・マサチューセッツ工科大学のメディアラボと共同開発し、無償で公開している教育用のビジュアルプログラミング言語 [5] で、60 以上の言語に対応しており、2021 年 10 月時点での全世界の登録ユーザーは 7900 万人以上である [6]（ユーザー登録無しに使うこともできるため、実際のユーザーはさらに多い）。8 歳から 16 歳をターゲットとして開発された言語であるが、米国・ハーバード大学の初級コンピュータサイエンスの授業でも Scratch が採用されおり、論文 “Scratch for budding computer scientists” にて、以下の実施結果が述べられている [7]。

Scratch は、学生のコンピュータサイエンスへの第一歩の時期に次の効果をもたらした。

- プログラミングへのモチベーションを向上させた
- プログラミングの基礎を習得させた
- プログラミングの初学者全員が Scratch の最初の経験がその後の Java の習得にプラスの影響があったと感じた

また、以下のような無償の教材作成がなされるなど、Scratch 言語はプログラミングの初歩を学ぶために有用で効果的な言語と位置づけられている。

- Creative Computing for Scratch 3.0[8]

  ハーバード大学教育大学院にて作成された K-12（日本の小中高校に相当）を教える教育者向けのカリキュラムガイド

  大学生への利用において「コンピュータを苦手とする文科系学生でも、楽しみながら創造的コンピューティングの概念を理解でき、実践できるようになります。難しい高級言語に取り組む前準備として、プログラミング入門編の講義に活用すると効果的です。」との記述がある。
- CS50 (Computer Science 50)[9]

ハーバード大学が公開している、オンラインのコンピュータサイエンス入門講座、2021 年 5 月には日本語翻訳版の「CS50.jp[10]」が公開された。Week0 から Week10 までの 11 週分の教材から成り、Week0 では Scratch が対象となっている。

Scratch はビジュアルプログラミング言語であり、マウス操作と簡単なキー入力だけでブロックをつなぎ合わせながらプログラムを組み立てることができる。ブロックにはつなぎ合わせるための凹凸がついており、つなげられるものだけがつながる仕組みになっているため構文上のミスは起こらない。一般のプログラミング言語は、テキストでプログラミングを行うため、入力ミスや構文上のミスに気を付けながらプログラムを作成する必要があり、初学者がプログラミングを学ぶ際に最も重要なプログラムの基本的な考え方やプログラムで実現したい内容の構築だけに集中することが難しい。

新しいプログラミング言語を学ぶ際に、「最初に書くべきプログラムは、”Hello World” を表示する」という慣習がある。プログラミング言語による記述の違いを図 2 に示す。この図からわかるように、Scratch が最もシンプルであり、直感的に作成することができる。その次に簡単なものは Python であり、数値演算や AI に関連する様々なライブラリが活用できることも加えて非常に人気の高いプログラミング言語である。データサイエンス入門 I の授業では、教育用の Scratch でプログラミングの楽しさに触れたのちに、実務で活用できる Python のプログラミング体験も行う。

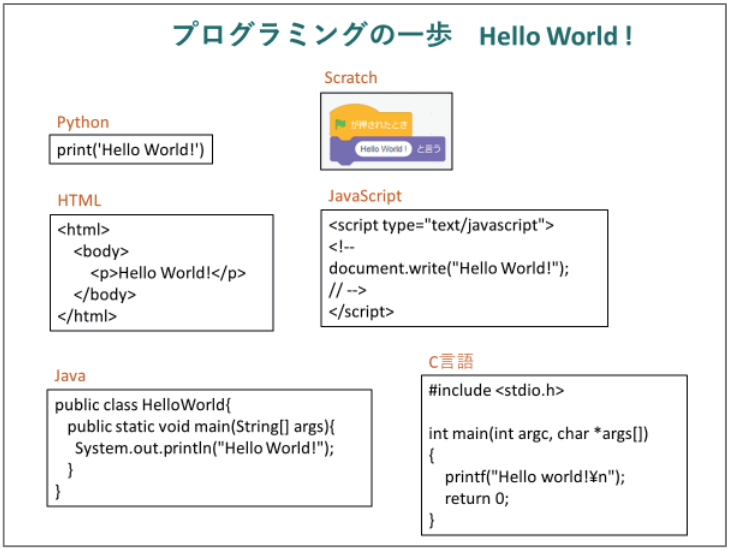

図 2.　各種プログラミング言語による “Hello World!” プログラム

4.　プログラミングの題材「音楽演奏プログラミング」と PDCA/DCAP サイクル

Scratch を使ってゲーム、アニメーションからアートまで、様々な作品を制作することができる。授業では、「音楽演奏プログラミング（一人オーケストラ）」を題材として、一人一人が好きな楽曲（楽譜）を選び、プログラムでその楽曲演奏を再現するというプログラミングに取り組むこととした。楽譜を見ながら、プログラムを作成していく作業の中で、プログラムの基本（順次・分岐・反復などのフローやメインルーチン・サブルーチン、変数、関数などの取り扱い、並列処理など）を身につけていくことができる。「音楽演奏」であるため、ブロックをつなげては音の確認をして小さな達成感を味わいながら完成へと進めていくことができる。

音符 1 つ 1 つを Scratch のブロックとしてつなげることでプログラムに仕上げることができるが、単につなげていくだけではなく、楽譜全体の中で繰り返しパ

ターンなどの規則性を発見して新しい関数を作ったり、複数の楽器を並列して演奏するフローを作成したり、独自の楽器の組み合わせやメロディ・リズムをアレンジしたり、音の強弱を工夫したりなどして、世界に 1 つだけの、自分だけのハーモニーを作り出すことができる。同じ楽譜を使う場合であっても、工夫により様々な異なる作品ができ上がるのである。プログラムの作り方によっては、プログラムの実行時に複数の楽器の音が徐々にずれてくる「音ズレ」の問題が発生する場合があるが、この原因を探り、うまく合わせるための工夫をするなど、試行錯誤も必要になる。このように、Scratch で音楽演奏プログラムを作成しながら、プログラミングの基本やデザインのプロセス、問題解決の取り組み方を学ぶことができる。こうして、思ったとおりの演奏ができたときには、大きな達成感を味わうことができるのである。

楽譜からプログラムを作成した例の一部を図 3 に示す。またこのプログラムの実行結果を以下のリンクにて紹介する。

https://drive.google.com/file/d/1Pp4bKISmQn2ouv6qMmM3OIxseO-eBx65/view?usp=sharing

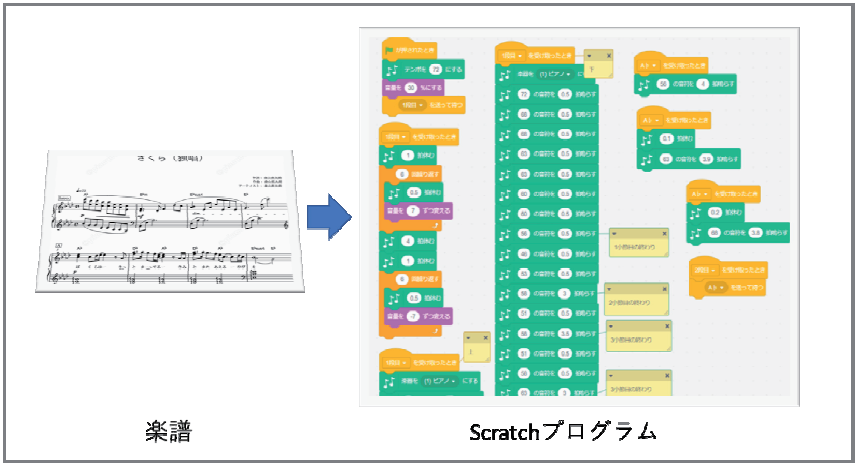

図 3. 楽譜から書き起こした Scratch プログラムの一部

このプログラム例では、楽譜にはない打楽器（トライアングル）やピアノ以外の楽器（ビブラフォン、スチールドラム）を追加するなどのアレンジやリタルダント (rit. : テンポを次第に落としてゆく表現) を含めている。

プロジェクトを推進する手法に「PDCA サイクル」という概念がある。Plan（計画）、Do（実行）、Check（評価）、Action（改善）の頭文字を取ったもので、品質管理をはじめ、目標達成のためのフレームワークとして様々な分野・場面で活用されている。プログラミング作業にもこれを当てはめることができる。Scratch プログラミングについて、初学者の場合と慣れてきた人の場合とに分けて考えてみることにする。初学者は、PDCA よりも、DCAP の方が当てはまりが良いと考えられる。DCAP は、Do（実行）、Check（評価）、Action（改善）、Plan（計画）、の順であり、PDCA とはサイクルの順番が異なる。行動をしながら試行錯誤を繰り返し、その末に計画を立てて、次の実行を進めるという考え方である。「習うより慣れろ」の方針で、まずは Scratch を使って楽譜の音符を 1 つ 1 つプログラムに落とし込みながら使い方に慣れ（実行）、正しくプログラミングできているかを逐次確認（評価）し、間違っていれば修正（改善）する、この作業を行う中で、楽譜の中の繰り返し箇所や何度も登場するフレーズに気づき、プログラムの可読性（読みやすさ、理解しやすさ）を向上する、効率化（関数化するなど）を図る、また、元の楽譜にアレンジを加えるなどの計画を練る、これを実行に移す、という流れができる。このように、DCAP のサイクルにより、素早くプログラミングに取り掛かりながら、質の高いプログラム作成へと進めていけることも Scratch の特長と言える。「新しいものが生まれては消えていく」現代においては、しっかり計画を立てて改善していく PDCA よりも、未知のものにまず飛び込んでみる DCAP のほうが有益だが、慣れ親しんだものであれば、PDCA が機能することも多い [11] と言われるように、Scratch プログラミングに慣れてきた場合には、まず楽譜を見てサブルーチンや関数を活用する箇所を見極めて、（他のプログラミング言語と同様に）計画的にプログラムを設計・作成していく PDCA 手法が望ましいと考えられる。

5. 学生のプログラム作品の事例

授業では、音楽演奏プログラミングの基本について解説し、簡単な実習を行った。詳細なプログラミングのヒントについては音楽演奏に特化した解説書を作成して配布し、学生にはこれを参考に新たな楽譜で作成したプログラム作品を提出することを課題とした。(実際には、本課題とロボットプログラミング、Excel によるデータ分析､の 3 種類の課題の中から 1 つを選択して提出することとしたが、Scratch による音楽演奏プログラムを提出する学生が全体の半数を超え、最も多かった。) プログラミングを楽しく体験しながらプログラムの基本を学習（楽習）することが目的であるため、プログラミング量の目安を楽譜全体でなく、1 ページ程度としていたが、提出作品の半数以上は 1 曲仕上げた形となっていた。また並列処理による厚みのある演奏、複数の楽器による独自のハーモニーの表現など音楽の作品としてクオリティの高い作品も多数あり、驚かされた。新型コロナ感染予防のために在宅自粛期間中であり、オンライン授業となっていたことも相まって､学生がプログラミングに集中し、満足できる作品制作に没頭したことが、作品から伝わってきた。プログラミングの基本のみならず、デザイン、問題解決など、様々な試行や工夫を重ねて作品作りに取り組んだことがプログラム作品から垣間見られた。

学生の承諾を得て、作品の一部を以下に紹介する。

①曲名：人生のメリーゴーランド

プログラム実行結果：https://drive.google.com/file/d/1yGGt6y-0FNoHsx1wkHPXCVGtWkCtUqri/view?usp=sharing

概要：10 種類もの楽器を使ってプログラミングされている。音の正確さはもとより、楽器ごとにメロディの中で音量の強弱や演奏速度が調整されている。全ての楽器のハーモニーがすばらしく、オーケストラ演奏が実現できている。

②曲名：生まれて初めて

プログラム実行結果：https://drive.google.com/file/d/1fLJwkf1-IL70XdqksDEI1R

KQ8z30xt4J/view?usp=sharing

概要：シンバルや馬のいななき、鳥のさえずりまで効果的に活用し、ミュージカルの一場面のような壮大な雰囲気が醸し出されている。

③曲名：Believer

プログラム実行結果：https://drive.google.com/file/d/1BKd718fwp54brxunGO7aI8wVSgAEe9Mp/view?usp=sharing

概要：打楽器を加えたり、音楽に抑揚をつけたりなど聞き飽きないような工夫がなされている。サブルーチンや関数を緻密に設計し、長い演奏が音ズレなく心地よくリズミカルに表現されている。

6. 学生の感想

Scratch による音楽演奏プログラミング実習に 2020-2021 年に取り組んだ学生の感想（抜粋）を表 1 に紹介する。また、感想の全てをテキストマイニングによりワードクラウド（文書の中で出現頻度が高く特徴的な単語をスコア化しスコアの大きさに応じた大きさでその単語を示すことによって、文書の内容をひと目で印象づけることができる図）にしたものを図 4 に示す。

テキストマイニングの結果から、サブルーチンを活用してプログラムを構築したこと、音がずれる問題に悩み工夫したことが見て取れる。また、「楽しい」と「難しい」の感想がほぼ同数であり、「最初は難しく思ったが実際に試してみると楽しく取り組めた」学生が多く、「難しい」だけの感想をもった学生は全体の 5%以下であることがわかった。特に印象的であったのは、「心を込めて」プログラムを作った（プログラムで心を込められると思わなかった）との感想である。Scratch で情操を育みながらプログラミングに取り組むことも可能であることを確信することができた。楽譜が読めなかった学生が、課題を通して読めるように取り組み、実際に読めるようになったとの話も聞き、プログラミングだけではない他の分野へも良い効果があることも確認した。

Scratch プログラミングを通して、プログラミングの基本、デザイン、問題解決を学ぶことができると記述したが、数学（音の拍数や音の大きさを細かく操作

する際に）も関係し、さらに情操も育むことができることを、改めて学生の感想から学ぶことができた。Scratch プログラミングはまさに STEAM 教育に適していると考えられる。

表 1．Scratch に取り組んだ学生の感想

Scratch プログラミング実習に関する大学生の感想　（抜粋 2020・2021 年度）
スクラッチを使ってプログラミングをするということを初めてやったので、**最初は興味もなく、難しそうだと思っていたが、実際にやってみたら思っていたより簡単で楽しかった。**
スクラッチを使ったプログラミングが面白かった。**楽譜を読み取って数字に起こしながら組み立てていくという作業が音楽を論理的に解釈しているようで、普段使うことが少ない脳の理数系の部分が刺激されている感じがした。**
テーマが音楽だったこともあり、何をすると何が起こるのかがいちばんよくわかりやすかった。プログラミングにはもともと興味があったが、今回の授業でさらにスクラッチをいじりたくなった。
音楽的な知識を数学的に理解するのが大変だ。つまり、音符があらわす音が続く長さを、0.25 のように、数字に変換できなければいけないということである。
授業でこんなに楽しいことができるとは思ってもいなかったので、本当に嬉しいです。**曲が作れたことで、自信が付きました。**
直感的にプログラミング体験ができるということに感動しました。
1 曲通して作ってみたいなと思いました。**もっと幼い頃から知りたかったです。**
Scratch はとても難しそうだなと感じていたが、楽譜さえ読めるようになればスムーズにできるだろうなと思った。**いい機会なので楽譜を読めるようになりたい。**
Scratch を利用して自分で音楽を作れるということに驚きました。しかし、音楽の知識がなければ少し難しくなってしまうと感じました。私は、楽譜を読むのが少し苦手なので、**頑張って楽譜を読めるように努力したいと思います。**
私は、世界に一つだけの花のサビ部分を作りました。まだ全然完成していないのですが、**授業でこんなに楽しいことができるとは思ってもいなかったので、本当に嬉しいです。**
Scratch では**音楽が得意でない自分でも演奏をすることができてとても楽しかったです。**
次は曲を作ります。と聞いた時は絶望しましたが、**思っていた何十倍も簡単でとてもとっつき易かったです。**
scratch での曲作りは自分にとって難しく感じられた。新しいもの作ることはとても難しく、**自分もクリエイティブな人間になれるよう、この授業についていきたいと思います。**
プログラムを作り始めると思っていたよりも熱中してしまい、楽しんで進めることができました。**楽しみながらプログラミングの基礎が学べるツールとして有効であることが実感できました。**
音楽を作成してみて率直に思ったのが**人の演奏にかなうものはない！**ということだ。ただ、**時間をかけて思いを込めて作って仕上がった作品としては感動的なものであった。**

図 4. 学生の感想のワードクラウド

7. おわりに

Scratch をプログラミング学習のきっかけとして取り入れ、「音楽演奏」を題材とすることで、楽しみながらプログラミングに取り組み、プログラミングの基本の習得、モチベーションの向上や、情操の育成などの効果があることを確認することができた。音楽演奏プログラムを提出課題とすることにより、質の高い作品を創作する学生が続出し、プログラムの構成内容のみならず、完成するまでの集中力や根気強さ、芸術性の高さにも驚かされた。

本課題は 2020 年度から開始し、半期ごとのカリキュラムであるため、2021 年度前期が 3 回目の実施であったが、期を追うごとに作品のレベルが上がってきていることにも気づいた。この理由としては、①学生に配布する説明資料を毎回バージョンアップしていること、②プログラミング実習を始める前に前回の学生の作品を紹介していること、が考えられる。特に②においては、「これから始めようとするプログラミングでこんな作品が作れる」という驚きや期待感を学生が持ち、モチベーションを向上させるとともに、もっと良い作品を作りたいという思いが、

さらに良い作品作りにつながっているのではないかと考える。

データサイエンス入門Ⅰでは、3種類のプログラミング実習（Scratch音楽演奏プログラミング、ロボットプログラミング、Pythonによる統計・画像処理プログラミング）を行っているが、最初にScratchプログラミングを実施することで、少なくとも「プログラミング」へのハードルを下げた状態で次の実習に取り組めていると感じる。Scratch学習がその後のプログラミング学習やその他の分野でどのような影響をもたらすかを定量的に評価しながら、さらに効果の高い授業内容に改善していきたいと考える。

謝辞

プログラム作品の実行結果の掲載を承諾いただきました。経済学部・小林美葵さん、文芸学部・木下佳音さん、社会イノベーション学部・長谷川七海さんに感謝申し上げます。

参考文献

[1] 内閣府 Society5.0 https://www8.cao.go.jp/cstp/society5_0/ （2021/10/30閲覧）
[2] Society 5.0に向けた人材育成 ～社会が変わる、学びが変わる～ https://www.mext.go.jp/component/a_menu/other/detail/__icsFiles/afieldfile/2018/06/06/1405844_002.pdf （2021/10/30閲覧）
[3] 文部科学省 STEAM教育等の各教科等横断的な学習の推進 https://www.mext.go.jp/a_menu/shotou/new-cs/mext_01592.html （2021/10/30閲覧）
[4] Society5.0に向けた人材育成について https://www.mext.go.jp/component/a_menu/education/detail/__icsFiles/afieldfile/2018/11/19/1411060_02_1.pdf （2021/10/30閲覧）
[5] Scratch https://scratch.mit.edu/about （2021/10/30閲覧）
[6] Scratch statistics https://scratch.mit.edu/statistics/ （2021/10/30閲覧）
[7] Malan, David J., and Henry H. Leitner. "Scratch for budding computer scientists." ACM Sigcse Bulletin 39.1 (2007): 223-227.
[8] Creative Computing for Scratch 3.0 (Japanese Version)

https://scratched.gse.harvard.edu/resources/creative-computing-scratch-30-japanese-version.html （2021/10/30 閲覧）
[9] CS50(Computer Science 50) https://cs50.harvard.edu （2021/10/30 閲覧）
[10] CS50.jp https://cs50.jp/ （2021/10/30 閲覧）
[11] DIAMOND online https://diamond.jp/articles/-/247269 （2021/10/30 閲覧）