

小学校算数の「正多角形」におけるロボットカーを用いた 試行錯誤を取り入れたプログラミング教育の試行

Developing and Trialing of Programming Lesson of “Regular Polygon” Introducing Trial-and-Error of Robot Car in Mathematics of Elementary School Education

(2020年3月31日受理)

佐々木 弘 記

Hironori Sasaki

Key words : プログラミング教育, コンピュータショナル・シンキング, プログラミング的思考, 試行錯誤

要 約

2020年度から小学校教育において実施されるプログラミング教育のねらいの一つである「プログラミング的思考」について、イングランドにおける教科「コンピューティング」の中核となる「コンピュータショナル・シンキング」との関連を整理した。また、小学校算数の「正多角形」を題材として、試行授業を開発し実践した。「正多角形」を題材とした試行は2018年にコンピュータだけを用いたプログラミングの学習活動を行っていたが、本研究では、更にミニコンピュータである「スタディーノ」と一体化したロボットカーの操作を取り入れた。紙面上に描かれた正多角形の線に沿ってロボットカーが動くようにプログラムする学習活動である。児童が実際にプログラミングによるロボットカーの操作を体験すれば、正多角形の性質やプログラミングの基本処理についての理解が深まると考えたからである。「子ども科学体験大学」の中で試行授業を実践し、アンケート調査等で検証したところ、実際にロボットカーを操作する活動を取り入れたことは、これらの理解にはあまり影響を与えないことが示唆された。

1. 研究の目的

2016年6月に、「小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」は、「小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）」を発表した¹⁾。それを受けて、2017年3月に告示された新しい学習指導要領では、小学校教育においてプログラミング教育を実施することが明示された。小学校にプログラミング教育が取り入れられるのは初めてのため、教師の不安を解消し、安心して取り組めるように、文部科学省は「プログラミング教育の手引」発刊しており、2020年2月には第三版が出ている（以下、「手引」と略す）。手引には、プログラミング教育のねらいを、次のように述べている²⁾。

①「プログラミング的思考」を育むこと、②プログラムの働きやよさ、情報社会がコンピュータ等の情報技術によって支えられていることなどに気付くことができるようにするとともに、コンピュータ等を上手に活用して身近な問題を解決したり、よりよい社会を築いたりしようとする態度を育むこと、③各教科等での学びをより確実なものとする

前出の有識者会議「議論のとりまとめ」においては、「プログラミング的思考」とは、「自分が意図する一連の活動を実現させるために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と説明している。また、

プログラミング的思考は「コンピューショナル・シンキングを踏まえている」と述べられている。

更に、手引きには、プログラミング的思考について次のように具体的に述べられている。

このことをコンピュータを動作させることに即して考えます。コンピュータに自分が考える動作をさせるためには、①コンピュータにどのような動きをさせたのかという自らの意図を明確にした上で、まず、②コンピュータにどのような動きをどのような順序でさせればよいのかを考えます。この際、意図した一連の動きが、一つ一つの動きをつなげたものであることを理解する必要があります。そして、③一つ一つの動きに対応する命令（記号）が必要であることを理解し、コンピュータが理解できる命令（記号）に置き換えた上で、④これらの命令（記号）をどのように組み合わせれば自分が考える動作を実現できるのかを考えます。さらに、⑤その命令（記号）の組合せをどのように改善すれば自分が考える動作により近づいていくのかということも試行錯誤しながら考えていきます。

一方、イングランドでは、2013年のナショナル・カリキュラムにおいて、教科「コンピューティング」が新設され、2014年9月より実施されている。この教科では、「コンピューショナル・シンキング」が中核をなす。そこで、次節において、イングランドにおけるコンピューショナル・シンキングについて紹介するとともに、我が国における「プログラミング的思考」との関連を整理する。その上で、学習指導要領に示された小学校算数におけるプログラミング教育の事例を参考にして、試行授業を開発し実践する。

ところで、独立行政法人国立青少年教育振興機構による「子どもゆめ基金」から2014年度以降継続して、「子ども学体験大学」をテーマとして助成を獲得している³⁾。その中でプログラミング教育に関する試行授業を実践し、効果を検証することを研究の目的とする。

2. イングランドにおけるプログラミング教育

2.1 目標

教科「コンピューティング」は、CS(Computer Science)、

IT(Information Technology)、DL(Digital Literacy)の3分野で構成されており、プログラミングは主にCSにおいて学習される。ナショナル・カリキュラムの“Purpose of study”では、目標を次のように述べている⁴⁾。

高度なコンピューティング教育は、コンピューショナル・シンキングを駆使し、世界を理解して変革する創造性を児童生徒に身に付ける。コンピューティングは、数学、科学、デザイン、テクノロジーに深くリンクしており、自然と人工の両方に見識を与えるものである。コンピューティングの中核は、コンピュータサイエンスである。ここでは、児童生徒は「情報や計算の原理」、「デジタルシステムがどのように動くか」、「プログラミングを通じてこの知識をどのように使うか」を習う。この知識と理解を土台に、児童生徒はプログラムやシステム、広範なコンテンツを創造するための情報技術を利用する力を身に付ける。コンピューティングは、確実に、未来の職場にふさわしいレベルで、そして、デジタル世界の積極的な参加者として、児童生徒がデジタル処理で読み書きできるようにICT技術を利用でき、それらを通して、自分の考えを表現し、アイデアを展開することができるように一する。

更に、“Aims”として、次のように述べている。

- ・教科「コンピューティング」のナショナル・カリキュラムは、児童生徒に次のことを習得させることをねらいとする。
- ・抽象概念、論理、アルゴリズム、データの表現を含むコンピュータサイエンスの基本原則と概念を理解し、応用することができる。(CS)
- ・コンピュータ用語で問題を分析でき、そのような問題を解決するために、コンピュータプログラムを記述する実践経験を繰り返すことができる。(CS)
- ・問題を分析的に解決するために、新しいまたはよく知らない技術を含む情報技術を評価し、応用できる。(IT)
- ・信頼でき、有能で、自信に満ちた、創造力のあるICTのユーザになる。(DL)

コンピューティングは一つの教科であり、3分野に分

かれているだけに、コンピュータやプログラミングそのものについての能力や技能の習得の色合いが強いと言える。

2.2 コンピュータ・シンキング

イングランドではComputing At School (CAS) が教科「コンピューティング」のための教員研修や教材・教具の提供を行っている。また、British Computer Society (BCS) が運営し、CASがサポートしているBarefoot Computing Projectでは、学校の教員向けに学習指導案や教材などがオンラインで提供されている（図1）⁵⁾。

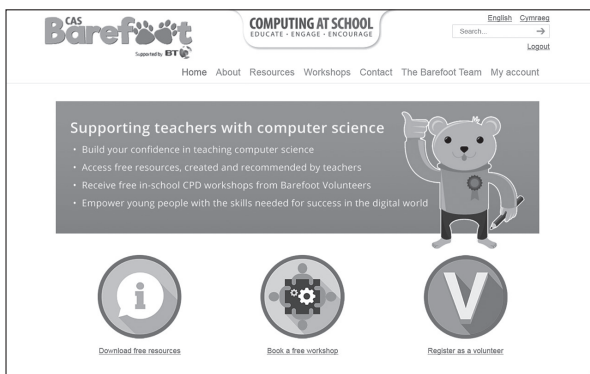


図1 BarefootのWebサイト

教員研修用の資料の中に、次のようにコンピュータ・シンキングについて解説したものがある。

コンピュータ・シンキングとは何か

コンピュータ・シンキングとは、コンピュータが問題解決を助けてくれる方法で問題を見ることだ。これには2つのステップがある。

1. 最初に、問題を解決するのに必要なステップを考える。
2. 次に問題についてコンピュータを動作させる技術を使う。

例えば、アニメーションを作るとき、ストーリーを考え、コンピュータのハードやソフトを使う前にどうやって撮影するか計画することから始めなければならない。コンピュータを使う前にやっておく思考が、コンピュータ・シンキングだ。

コンピュータ・シンキングは、コンピュータについての思考や、コンピューのような思考ではない。コンピュータはまだ自分で思考しない。少なくとも

も今のところは。

コンピュータ・シンキングには、6つの概念と5つのアプローチがある

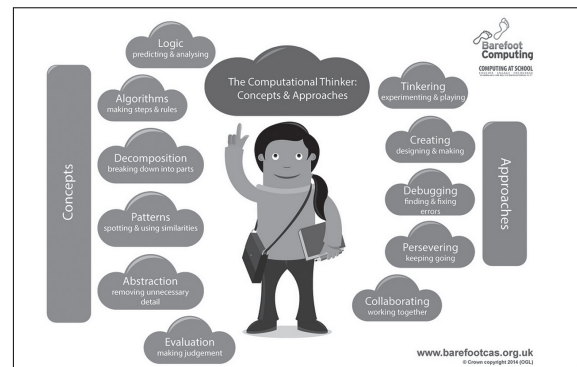
○概念

- ・ロジック／予想し、分析する
- ・アルゴリズム／ステップとルールを作る
- ・分割／部分に分ける
- ・パターン／位置決めと類似性を使う
- ・抽象／必要のない細部を除く
- ・評価／判断する

○アプローチ

- ・試行錯誤／試しにやってみる
- ・創造する／デザインして作る
- ・デバッグする／エラーを見つけて修正する
- ・忍耐強く／やり続ける

協働する／一緒に活動する



なぜ、コンピュータ・シンキングは、大切なのか

コンピュータ・シンキングとその背後のコンセプトは、コンピュータサイエンスからの多くの基礎に基づいている。コンピュータ科学者は、問題解決の最も効率的な方法を探すことに興味がある。彼らは、問題を正しく、最も速く、しかも最小のリソース（時間やスペース）を使って解決するベストの方法を見つけないのだ。

先に示した我が国の「プログラミング的思考」に比べると、イングランドの「コンピュータ・シンキ

ング」の概念はかなり幅広く、多岐にわたるものである。従って、プログラミング的思考は、コンピューショナル・シンキングの概念の一部に包含される関係にあると言える。

ところで、両者に見られるキーワードとして、「試行錯誤」が挙げられる。試行錯誤について、手引には、算数において「正多角形をかく」場合を例示しながら、次のように述べられている。

このように、児童は試行錯誤を繰り返しながら自分が考える動作の実現を目指しますが、思い付きや当てずっぽうで命令の組合せを変えるのではなく、うまくいかなかった場合には、どこが間違っていたのかを考え、修正や改善を行い、その結果を確かめるなど、論理的に考えさせることが大切です。

つまり、児童は見通しをもって自分のつくったプログラムの命令を修正し、実際に動作させたときとの結果を比較することが大切になってくる。したがって、試行授業を設計する際に、児童の試行錯誤を支援する方法を工夫することが必要となる。

3. 試行授業の開発

3.1 題材

手引には、プログラミングに関する学習活動例が分類して提示されている。その中で「A 学習指導要領に例示されている単元等で実施するもの」として、学習指導要領の、小学校第5学年の算数において、正多角形の作図を行う学習が上げられている。

図形を構成する要素に着目し、プログラミングを通した正多角形のかき方を発展的に考察したり、図形の性質を見いだしたりして、その性質を筋道を立てて考え説明したりする力を確実に育みます。

プログラミングを通して、正多角形の性質を基に正多角形をかく場面である。そこで、本研究において行う試行授業の題材として、この「正多角形」の題材を扱うこととした。

3.2 ねらい

プログラミング教育の目的には、算数での学びをより

確実なものとするのが含まれているので、算数の目標の達成が第一となる。学習指導要領の第5学年「B 図形 B(1) 平面図形の性質」には、次のように記されている⁶⁾。

- (1) 平面図形に関わる数学的活動を通して、次の事項を身に付けることができるよう指導する。
- ア 次のような知識及び技能を身に付けること。
- (ア) 図形の形や大きさが決まる要素について理解するとともに、図形の合同について理解すること。
- (イ) 三角形や四角形など多角形についての簡単な性質を理解すること。
- (ウ) 円と関連させて正多角形の基本的な性質を知ること。
- (エ) 円周率の意味について理解し、それを用いること。
- イ 次のような思考力、判断力、表現力等を身に付けること。
- (ア) 図形を構成する要素及び図形間の関係に着目し、構成の仕方を考察したり、図形の性質を見だし、その性質を筋道を立てて考え説明したりすること。

「正多角形」の題材では、(1)のアの「(イ) 三角形や四角形など多角形についての簡単な性質を理解すること」がねらいとなる。プログラミング的思考と関連させ、「試行錯誤しながら曲がる角度と正多角形の一つの内角の大きさとの関係を見いだすこと」を一つのねらいとする。

更に、手引には、次のような学習活動が示されている。

児童は、手書きで正方形を作図する際の「長さ□cmの線を引く」、「(線の端から) 角度が90度の向きを見付ける」といった動きに、どの命令が対応し、それらをどのような順序で組み合わせればよいのかを考え(プログラミング的思考)、また、繰り返しの命令を用いるとプログラムが簡潔に書けることに気付いていきます。

自分の意図した動きに対応した命令を組み合わせ、順番に上から実行されることと、繰り返し命令により効率

的なプログラム作成のよさに気付くことをねらいとしていることから、「命令が上から下へと順番に実行される『順次』を理解した上で、繰り返しの命令を用いる『反復』のよさを見いだすこと」を二つ目のねらいとする。

①正多角形の一つの内角の大きさ

試行錯誤しながら、曲がる角度と正多角形の一つの内角の大きさとの関係を見いだすこと。

②基本処理の「順次」と「反復」

命令が上から下へと順番に実行される「順次」を理解した上で、繰り返し「反復」のよさを見いだすこと。

したがって、これら二つのねらいを達成するための指導方法を工夫することとした。

3.3 指導方法の工夫

筆者は、特定非営利活動法人みんなのコードが運営しているサイト・プログラムの「多角形コース」を利用し、2018年3月にプログラミング教育を試行した。「多角形コース」では、プログラミングでキャラクタを動かして図形をかきながら、多角形の性質を学習できるコースとなっており、操作に慣れた後、繰り返しブロックなどの命令を使って正方形・正三角形といった図形をかくことができる。また、かきたい図形を自由にかけけるステージも用意されている(図2)⁷⁾。



図2 プログル「多角形コース」

この試行授業には、小学生28名が参加し、一定の効果を上げることができた⁸⁾。ところが、キャラクタがコンピュータの画面上で正多角形をかくだけの動作なので、見通しをもって試行錯誤するには不十分であった。やはり、実際にプログラムでモノを動かして確かめないと、

試行錯誤とは言えないと考えた。

そこで、本研究においてはアーテック社の「スタディーノ」⁹⁾を用いて、コンピュータのプログラムにより実際にモノを操作し、試行錯誤する活動を取り入れようように考えた。スタディーノとは、ミニコンピュータのことで、コンピュータ画面上でスクラッチによるプログラムができるソフトウェアが付属している。スクラッチで作成したプログラムをケーブルでスタディーノに転送することができる。更に、スタディーノには、モーターや車輪を取り付けることができるので、ロボットカーを組み立て、転送されたプログラムに応じて、ロボットカーに接続されたモーターを回転させることができる。また、各種センサーを取り付けることができるので、接続されたセンサーから情報を収集し、動作をコントロールすることもできる。コンピュータの画面上でのキャラクタの動きと、ロボットカーの動きを対応させて学習することで、より学びが深まるのではないかと考えたからである。ここでは、スタディーノにモーターと車輪を接続してロボットカーを組み立て、紙面上に描かれた正多角形の線に沿って動くように試行錯誤しながらプログラムする学習活動を取り入れる。

①正多角形の一つの内角の大きさ

まず、コンピュータの画面上でキャラクタが正多角形をかくプログラミング活動から始める。最初は、正四角形(正方形)で、児童は「90度曲がる」とプログラムすればよいことは容易に理解できると考えられる。次に正三角形をかくときに、一つの内角が60度なので、「60度曲がる」とプログラムすると考えられるが、それでは正三角形はかけない。そこで、180度から60度を減じ、120度にしなければならないことを見いださせる。さらに、正五角形へと進んだ際に、作図して最後に元の位置に戻ってくるためには、5回曲がって1回転(360度)しなければならないので、曲がる角度が $360 \div 5 = 72$ 度となることを見いだすことができるようにする。

また、実際に組み立てたロボットカーを操作するプログラムをつくり、紙面上に描かれた正多角形の線に沿って動くように試行錯誤しながらプログラムする学習活動を行うことで、プログラムでかいたどの命令がロボットカーのどの動作に対応しているのかを実感できるようにする。

②基本処理の「順次」と「反復」

プログラムの命令が上から下へと順番に実行されることをプログラムしながら確認する。正三角形から正六角形まで「順次」で記述させ、同じ命令を何度もプログラムしていることに気付かせる。その上で、「反復」を使えば効率的であることを見いださせる。

また、「反復」を使ったプログラムでロボットカーを操作し、同じ動作を繰り返しながらロボットカーが紙面上の正多角形の線に沿って動いていることを実感できるようにする。

3.4 評価の方法

①正多角形の一つの内角の大きさ


正多角形の一つの内角の大きさを求める質問紙(図3)による調査を2018年の調査結果と比較して検証する。

アンケート

※このアンケートの回答は、大学の研究だけに使うもので、学校の成績などには、まったく関係がありません。


1 次の正多角形の一つの角の大きさを教えてください。

(1) 正三角形




_____度

(3) 正五角形




_____度

(2) 正四角形(正方形)



_____度

(4) 正六角形



_____度

図3 内角の大きさを求める質問紙

②基本処理の「順次」と「反復」

「順次」と「反復」に関する、5段階尺度法によるアンケート調査(図4)を2018年の調査結果と比較して検証する。

	1	2	3	4	5
(問) 次の問いについて、あなたの気持ちにもっとも近い番号に○をつけてください。	そう 思わ ない	あ ま り そ う 思 わ ない	ど ち ら と も 思 え ない	や や そ う 思 う	そ う 思 う
(1) プログラムでは、命令が上から下へと「順番に実行される(順次)」ことが理解できたと思います。(知・理)	1	2	3	4	5
(2) 「順次」を使って多角形をかくプログラムを作れると思います。(技能)	1	2	3	4	5
(3) プログラムには、「繰り返し(反復)」があることが理解できました。(知・理)	1	2	3	4	5
(4) 「繰り返し(反復)」を使って多角形をかくプログラムを作れると思います。	1	2	3	4	5

図4 アンケート調査

4. 試行授業の実践

4.1 日時、会場、参加者

- ・日 時：2019年3月3日(日) 10:00から11:30
- ・会 場：A大学内講義室
- ・参加者：小学生13名(4年生3名, 5年生6名, 6年生4名)

4.2 実践の様子

まず、試行授業の流れを説明した後、正多角形の性質について復習をした。次に、ノートパソコンで付属しているスクラッチのソフトウェアを起動し、コンピュータの画面上でキャラクターが正多角形をえがくプログラミング活動から始めた。児童は、スクラッチでの命令のかき方を学習した後、実際にプログラミングを開始した。はじめは正四角形(正方形)からである(図5)。

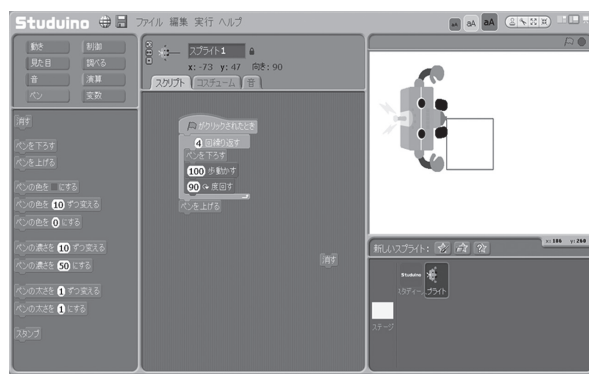


図5 付属のソフトウェア

予想通り、児童は「90度曲がる」とプログラムすればよいことは容易に理解できた。次に、正五角形をかくときに、何度曲がればよいかを考え、元の場所に戻ったときには、1周(360度)することをヒントとした。何人かは、360度を角の数で割ればよいことを見いだした。正三角

形から正六角形まで「順次」で記述させると、児童は同じ命令を何度も繰り返していることに気付いた。そして、「反復」へと進み、この方が効率的であることを見いだした。その後、作成したプログラムをスタディーノに転送し、ロボットカーを紙面上に描かれた正多角形の線に沿って動作される活動を行った（写真1）。



写真1 試行授業の様子

5. 結果と考察

①正多角形の一つの内角の大きさ

正三角形から正六角形の一つの内角の大きさを求める問題（全4問）の正答数を2018年の結果と比較したものを表1に示す。正答数の平均値は2018年が3.00、本試行では、3.15と向上しているが、統計的な有意差（5%）は見られなかった。従って、実際にロボットカーを操作するために試行錯誤しながらプログラムする活動を取り入れたことは、正多角形の内角の大きさの理解にはあまり影響を与えないことが示唆された。

表1 正多角形の内角の正答数（全4問）の比較

	2018年		2019年		t検定	
	m	SD	m	SD	t値	判定
正答数 （全4問）	3.00	1.10	3.15	1.41	0.44	
	n = 28		n = 13		*:p<0.05	

②「順次」と「反復」

アンケートの平均値(m)と標準偏差(SD)を2018年と比較したものを表2に示す。いずれの質問項目の平均値も2018年より本試行の方が高くなっているが、統計的な有意差（5%）は見られなかった。従って、実際にロボッ

トカーを操作するために試行錯誤しながらプログラムする活動を取り入れたことは、プログラミングの基本処理の理解にはあまり影響を与えないことが示唆された。

表2 アンケート結果の比較

番号	質問項目	2018年		2019年		t検定	
		m	SD	m	SD	t値	判定
1	プログラムでは、命令が上から下へと「順番に実行される（順次）」ことが理解できたと思いますか。	4.57	0.84	4.69	0.85	0.428	
2	「順次」を使って多角形をかくプログラムを作れると思いますか。	4.14	1.24	4.46	0.66	0.868	
3	プログラムには、「繰り返し（反復）」があることが理解できたと思いますか。	4.39	1.10	4.69	0.63	0.911	
4	「繰り返し（反復）」を使って多角形をかくプログラムを作れると思いますか。	4.29	1.12	4.69	0.63	1.22	
平均		4.35		4.63			
		n = 28		n = 13		*:p<0.05	

6. ま と め

プログラミング教育のねらいの一つである「プログラミング的思考」について、イングランドにおける教科コンピューティングの中核となる「コンピューショナル・シンキング」との関係を整理した。イングランドの「コンピューショナル・シンキング」の概念はかなり幅広く、多岐にわたるものであり、プログラミング的思考は、コンピューショナル・シンキングの概念の一部に包含される関係にあることを指摘した。

また、小学校算数の「正多角形」を題材として、試行授業を開発し実践した。本研究では、ミニコンピュータである「スタディーノ」と一体化したロボットカーの操作を取り入れた。実際にプログラムでロボットカーを動かして確かめることで、試行錯誤が実現すると考えたからである。2019年3月に「子ども科学体験大学」の中で試行授業を実践し、アンケート調査等で検証したところ、実際にロボットカーを操作するために試行錯誤しながらプログラムする活動を取り入れたことは、試行授業のねらいとした正多角形の内角の大きさの理解やプログラミングの基本処理の理解には、あまり影響を与えないことが示唆された。

今後も「プログラミング的思考」を育成するための指導方法を他の教科や題材についても工夫し、試行授業の実践を通して効果を検証していきたい。

謝辞：本研究において実践した「子ども科学体験大学」は、独立行政法人国立青少年教育振興機構が募集する2018年度子どもゆめ基金助成金を受けた活動である。ここに記して謝意を表す。

付記：本稿の一部は、日本科学教育学会研究会報告¹⁰⁾に大幅に加筆修正を加えたものである。

参 考 文 献

- 1) 文部科学省：小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ），2016
http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm
- 2) 文部科学省：小学校プログラミング教育の手引（第三版），2020
https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf
- 3) 独立行政法人国立青少年教育振興機構：子どもゆめ基金，2020 <https://yumekikin.niye.go.jp/index.html>
- 4) National curriculum in England: computing programmes of study :
[https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes -of-study](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study)
- 5) Barefoot Computing Project : <https://barefootcas.org.uk/>
- 6) 文部科学省：小学校学習指導要領解説算数編，2018
https://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afielddfile/2019/03/18/1387017_004.pdf
- 7) プログル：<https://proguru.jp>
- 8) 日本科学教育学会：研究会研究報告32(9)，15-18，2018
- 9) スタディーノ：<https://www.artec-kk.co.jp/studiuno/ja/>
- 10) 前掲8)