

# オンラインプログラミング学習環境 BitArrow を用いた JavaScript の授業実践報告

間辺 広樹<sup>1</sup> 長島 和平<sup>2</sup> 並木 美太郎<sup>2</sup> 長 慎也<sup>3</sup> 兼宗 進<sup>4</sup>

## 概要：

高等学校情報科のプログラミング入門の授業において，JavaScript の学習をオンライン学習環境 BitArrow を用いて実践した．4 回の授業の様子とテスト及びアンケートの結果から，BitArrow の有用性を評価する．その上で，基本制御構造の理解を目的として実践した授業内容を今後のプログラミング教育の一手法として提案する．

**キーワード：** BitArrow, プログラミング教育, JavaScript, ドリトル, 情報科学教育

## Report of JavaScript Lessons on 'BitArrow', Online Programming Learning Environment

MANABE HIROKI<sup>1</sup> NAGASHIMA KAZUHEI<sup>2</sup> MITAROU NAMIKI<sup>2</sup> CHO SHINYA<sup>3</sup> KANEMUNE SUSUMU<sup>4</sup>

**Abstract:** This paper provides a report of programming lessons at high school by using 'BitArrow', online programming learning environment. Through the observation of the students at classes, results of test and questionnaire survey, we evaluated BitArrow. Additionally, we suggest the lesson plans for understanding of basic control structure as a programming education

**Keywords:** BitArrow, Programming Education, JavaScript, Dolittle, Informatics

## 1. はじめに

情報社会の進展に伴って，誰もが情報機器や情報サービスを日常的に利用するようになった．しかし，これらの機器やサービスがプログラムによって動いていることを意識する利用者は少ない．車を運転するに際して，その動作原理や交通ルールを知らなければいけないことと同様に，本来情報機器や情報サービスもその仕組みを理解することで，安全かつ有効に利用できるようになるはずである．

その意味で公教育の果たすべき役割は大きく，どの校種でも情報教育にかける比重が大きくなってきた．高等学校では，平成 15 年に共通教科情報科が新設された．その後，

学習指導要領の改訂も経て，現在は「社会と情報」「情報の科学」の 2 科目が設置されていて，学校または生徒個人の判断でどちらかを選択することが決められている．科目「情報の科学」では，単元「問題の解決と処理手順の自動化」においてプログラミングを学習することが想定されている．

更に次期改訂を見据えているであろう中央教育審議会の第 197 号答申 [1] では，『共通必修科目として，問題の発見・解決に向けて，事象を情報とその結び付きとして捉え，情報技術を適切かつ効果的に活用する力を全ての生徒に育む「情報 I」を設定．全ての高校生がプログラミングによりコンピュータを活用する力を身に付けられるようにする．』として全ての高校生がプログラミングを学ぶことを想定している．

このようにプログラミング教育に対する社会的な要求は

<sup>1</sup> 神奈川県立柏陽高等学校

<sup>2</sup> 東京農工大学

<sup>3</sup> 明星大学

<sup>4</sup> 大阪電気通信大学

増しているが、学校現場では何を使ってどのように教えればいいのか、に対する明確な答えがない。共通教科情報科の教科書においても、使われているプログラミング言語には複数の種類があり、指導する教員の考え方も影響するため、授業方法は多様である。

本研究では高校生に対するプログラミング教育に用いる言語として、JavaScript を提案する。同言語が Web を中心に広く使われていることや、複数の教科書で採用されていることが主な理由である。しかし、一般的な JavaScript の学習法は操作やエラー表示の点などに課題があり、初学者が躓く要因となっていた。本研究ではそれらの課題を解決したオンラインプログラミング学習環境 BitArrow を用いて、教科書に準拠した内容の授業を実践した。生徒の観察、テスト、アンケートの結果から、BitArrow の有用性を評価すると共に、実践した学習法をプログラミング教育の一手法として提案する。

## 2. 研究の背景

### 2.1 プログラミング教育の課題

プログラミングはコンピュータに手順を教える作業であるが、日常的に行う思考とは異なるため、学習の初期段階でプログラミング嫌いを作らないための授業法が必要である。一般的に、教育用プログラミング言語は、ビジュアル型言語とテキスト記述言語に分けられるが、高等学校においては本格的なプログラミング学習への接続を想定して、英語表記によるテキスト記述言語を選択することが望ましいと考えられる。実際、共通教科情報科で利用されている複数の教科書では、VBA や JavaScript などのテキスト記述言語が使われている。

本研究では JavaScript に着目する。JavaScript が教育用あるいは入門用言語として使われる背景には、テキストエディタと Web ブラウザがあれば実習できるため、特別なソフトをインストールする必要がないことや、スクリプト言語であることからコンパイルして実行といった一連の手続きを行わずに学習できるといった手軽さがある。また、ゲーム等にも広く使われているため、多くの生徒が利用した経験があると共に、Ajax(Asynchronous JavaScript + XML) に象徴されるように、他の技術と組み合わせられて先端の Web 技術を形成しているため、学習したことを応用して発展的に活用していくことも期待できる。

ただし、JavaScript をプログラミング学習の導入とすることには課題がある。教科書や入門書、実践事例 [2] など、一般的な JavaScript の利用法は、次のような手順でプログラムを作成することが記されている。

- (1) テキストエディタでプログラムを入力する。
- (2) プログラムは「html」を拡張子として保存する。
- (3) Web ブラウザにそのファイルを読み込み、動作を確認する。

- (4) 修正や更新は、テキストエディタでプログラムを書き換え、上書き保存した後、Web ブラウザでファイルの読み込みをし直す。

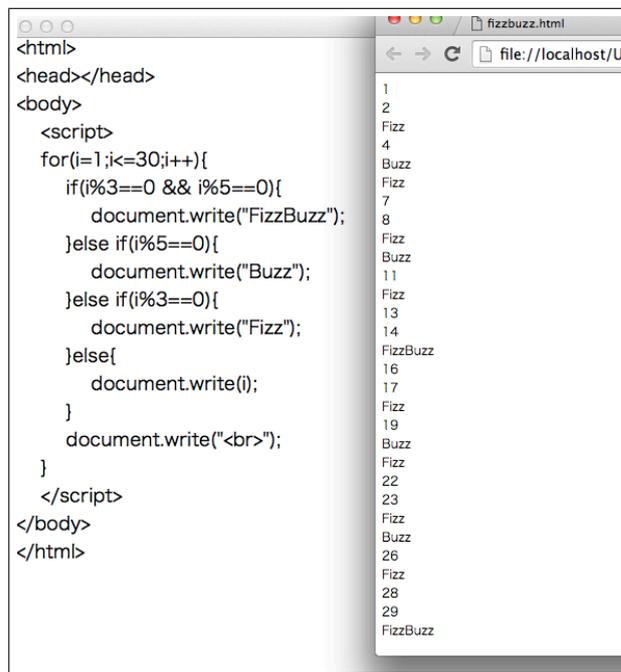


図 1 エディタとブラウザによる従来型 JavaScript 学習法

しかし、筆者らはこれまでの授業実践 [3] から、パソコンの苦手な生徒に習得させるには、これら数行の操作であっても、指導に困難が伴うことを体験してきた。

まず、この環境ではプログラムにエラーがあってもそれが表示されず (図 2)、何も表示されない Web ブラウザを見て途方に暮れる生徒が多かった。また、生徒はテキストエディタ、拡張子、Web ブラウザなどの用語を理解していないため、それぞれ用語についての説明や実習が必要であった。特に、ファイル操作に躓く生徒は、指定した場所に保存できなかつたり、拡張子を全角文字で入力してしまうことが多かった。プログラミングが始まると、テキストエディタと Web ブラウザというウィンドウを常に切り替えながら操作する必要があるため、それに伴うトラブルもあった。例えば、Web ブラウザの再読み込みをファンクションキーの F5 で行っている生徒が、テキストエディタ上で同じ操作を行ったとき、その時の日付や時間が表示されてしまうというトラブルなどもあった。更に、高等学校の情報科教育では、授業時間数は通常 1 週間に 1 回ないしは 2 回と少ないため、生徒が操作に慣れるまで授業の度に操作説明と個別の対応が必要であった。授業ではトラブルの生じた生徒に対して教師が個別の対応を強いられることになるため、本来のプログラミングの学習に入るまでかなりの授業時間と生徒・教員の労力を消費した。

このような授業実施上の課題を少なくし、生徒を本来のプログラミング学習に集中させるためには、次のような環

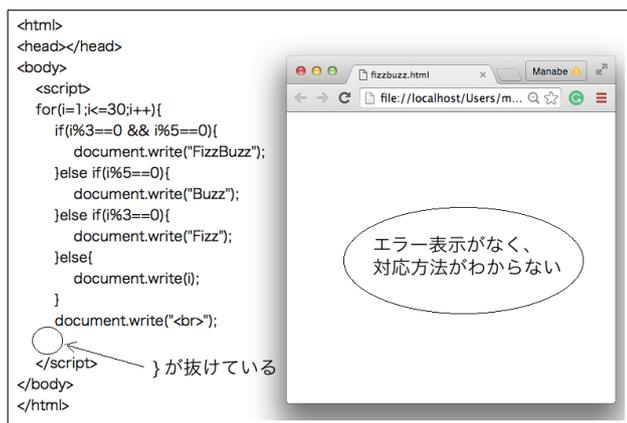


図 2 ミスがあってもわからない

境で授業をすることが望ましい。

- (1) 実行時のエラーとその発生箇所をわかりやすく表示する
- (2) ファイル操作に関するトラブルを回避する
- (3) ウィンドウ操作に関するトラブルを回避する

本研究では高等学校で科目「情報の科学」を学ぶ1年生を対象に、これらの課題への対応が考慮されたオンラインプログラミング学習環境 BitArrow を用いて、従来の課題が解決できるかどうかを調査する。その上で、同環境を用いて情報科の教科書に記された内容に準拠した学習を行った際に得られる学習効果と学習態度を調査した上で、実践した内容と方法をこれからのプログラミング教育の一手法として提案する。

## 2.2 BitArrow の概要

BitArrow (旧称: JSLesson) は長島らによって開発が進められてきたオンラインプログラミング学習環境である [4]。現在、JavaScript, C 言語, ドリトルの3言語から選択して利用することができる。サーバー上で動作するため、教員は生徒毎にプログラムの実行ログを含めた利用状況を掌握し、理解度の掌握や個別のアドバイスなど手厚い指導に繋げることも可能である。また、<img> などの要素にアクセスする DOM(Document Object Model) を操作するライブラリが用意されているため、スペルミスを引き起こしやすい長い命令を短くしたり、プログラムの実行を一時的に停止させるなどの難しい処理を容易に実現できるようになっている。

JavaScript は一般に Web ページを構成する HTML ファイルの script タグ内に記述するが、BitArrow には HTML と JavaScript のタブが用意されていて、これを使い分けることで2通りの学習方法から選ぶことができる。

図 3 の方法は、HTML タブ内だけを用いて従来型の JavaScript のプログラムを BitArrow のにプログラムを記述して、実行するものである。従来のエディタとブラウザを使い分ける課題を解消していることと、一般的な JavaScript

のプログラムを利用できるというメリットがある。

図 4 の方法では、HTML タブ内に要素名を定義するなど必要なことを記述した上で、JavaScript タブ内にプログラム本体を分けて記述して、実行するものである。こちらの方法では通常の命令が BitArrow 固有の命令に置き換わる (例えば document.write が setText になる、など) が、前述した DOM を利用できるというメリットがある。



図 3 BitArrow 版 JavaScript の画面



図 4 DOM を利用できる BitArrow 固有の画面

テキストエディタと Web ブラウザの組合せ (図 1) では、前章で記した学習上の課題が生じるが、BitArrow はそれらの課題を解消している。例えば従来ではエラーメッセージが表示されなかった (図 2) という課題は、図 5 のように解決され、学習者がその要因を特定しやすくなった。

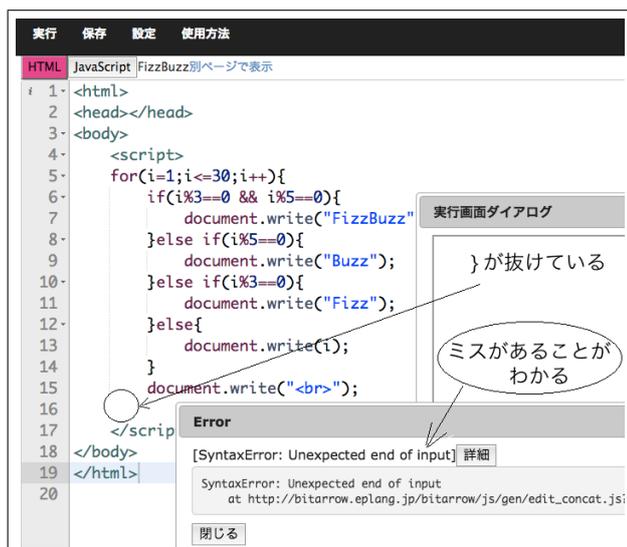


図 5 エラーメッセージでミスを教えてくれる

表 1 プログラミング教育の授業計画 (各 65 分)

言語	授業の内容と目標	主な学習課題
1 時間目	ドリトル 順次の理解	お絵描き
2 時間目	JavaScript 条件分岐の理解	奇偶判定, じゃんけん
3 時間目	JavaScript 反復構造の理解	画像表示, FizzBuzz 体験
4 時間目	JavaScript 自習 (復習と課題)	FizzBuzz, コラッツの予想
5 時間目	JavaScript テスト	アンケートフォームに入力

### 3. 実証の方法

#### 3.1 評価方法と学習者情報

本研究では BitArrow を用いて高校 1 年生 154 人を対象に行う。研究に先立って、生徒にパソコンの利用状況とプログラミングへの関心の度合いなどをアンケート調査した。生徒のほぼ 97.7% がスマートフォン等の携帯端末を常用しているが、パソコンを利用する機会を多く持たず、全体の 77.9% の生徒がパソコンの利用頻度が、1 ヶ月に数日程度かそれ以下のため、73.0% の生徒がパソコン操作に苦手意識を持っていた。その一方で、全体の 97.4% の生徒がプログラミングは「難しそう」と感じている反面、89.6% ができるようになりたいと思っていると回答した。これらのことから、生徒はプログラミングの難しさは覚悟しているが、その必要性を認識しているため、学習を受け入れる下地はある。ただし、操作スキルが不足しているため、それを考慮した授業法が必要である、と言えた。

授業は、プログラムの基本的な制御構造とその利用法を理解させることを目標とし、1 コマ 65 分の授業を週に 1 回ずつの合計 4 時間を行った後、5 時間目にテストを行う (表 1)。

以下が、評価のために設定した 3 観点である。

- (1) 学習中の生徒の様子を観察 (操作状況と学習態度)
- (2) 毎時間の「難しさ」と「楽しさ」の授業アンケート
- (3) 学習内容の理解度テスト

授業内容と調査の内容と方法については、以下に詳細を述べる。

#### 3.2 授業毎の内容

各授業内容の特徴を示す。1 時間目は、生徒の多くはプログラミング経験がないため、日本語で記述できるドリトル [5] を導入として利用することとした。題材は「歩く」や「右 (左) 回り」という単純な命令を並べるだけのお絵描きをするものである (図 6)。これによって、プログラムは上から順に実行されるという順次処理と、単純な命令でもその組合せによって様々な処理ができる (お絵描きできる) ことを理解させることを目的とした。

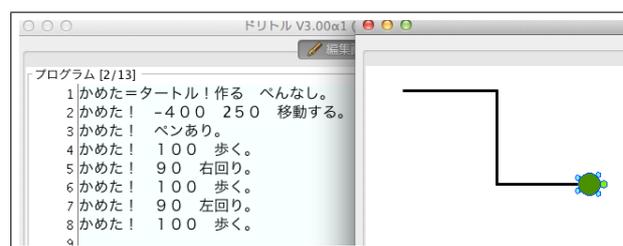


図 6 1 時間目・ドリトルによる描画プログラムの例

2 時間目は、前半で HTML の基礎知識を習得した後、後半で JavaScript の使い方と条件によって処理を分ける分岐を理解させることを目標とした。まず、HTML では背景色が文字で設定できることを実際に確認させたり、既存の Web ページが HTML で作られていることを示す。その上で、JavaScript が HTML に埋め込まれて使われることを示した上で、乱数を使えることと、乱数で発生させた整数が奇数か偶数かを判断させるプログラムを説明する (図 7)。これは被験者が使用している教科書にも出ている題材であり、応用させることでコンピュータとじゃんけんをするなど他のプログラムに発展させられることも理解させることとした。



図 7 2 時間目・JavaScript によるジャンケンプログラムの例

3 時間目は、繰り返しの構造を持たせることによって、同種の処理がまとめられることを理解させることを目標とし

た。そのために、同一の画像を複数個表示させる題材を扱う(図8)。また、4時間目に条件分岐と組み合わせることで複雑な処理が可能になることを理解させることを見据え、生徒を2人1組のグループにして、1から順に数を数えていきながら、「3で割り切れるときはFizz, 5で割り切れるときはBuzz, 3でも5でも割り切れる時はFizzBuzz」と言う活動を行わせる。その上で、これがFizzBuzz問題という古典的な課題であることと、「分岐」と「繰り返し」の組合せによってできることを説明し、途中までプログラムの作り方を示しながら、残りの部分を生徒に作成するよう指示する。



図8 2時間目・JavaScriptによる画像表示プログラムの例

4時間目は、3時間目に行ったFizzBuzz問題の模範解答を示した上(図4)で、自習課題として、コラッツの予想問題やグラフィカルな画像表示の問題などを説明したプリントを配布し、自由に作らせることとする。

5時間目は、それまでの学習がどの程度理解できているかを計るために、テストを実施することとした。問題を示しその処理を行うプログラムをアンケートのフォームから入力させるものである。つまり、プログラムを実行せずにプログラムを記述させ、複数の観点をチェックして、何を理解でき、何を理解できなかったのかを計るものである。

### 3.3 調査の内容と方法

調査する内容は、以下の通りである。

**調査1** 授業時の生徒の様子

**調査2** 学習内容の理解

**調査3** 授業の難しさと楽しさ／自由記述

調査1では、授業中の生徒の様子を観察し、従来型のテキストエディタとブラウザを使う学習法の課題が、BitArrowを使うことによって解決するかどうか、という『従来型の学習法の課題』の有無を調査する。また、DOMの使用が生徒の学習態度に影響があるかどうか、という『DOM利用による学習態度』についても観察する。

調査2では、プログラミングの基本である制御構造に関する2題のテスト問題を示し、実行せずにプログラムを考えて、アンケートフォームに入力させ採点する。問題は以下の2題を用意した。

#### 問題(1)

次のように画像を表示させるプログラムを作りなさい。

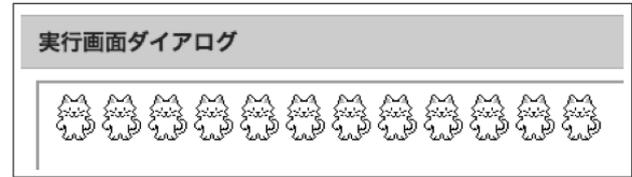


図9 問題(1)

#### 問題(2)

変数Aに0から99までの乱数を代入をする(A=rnd(100);)。Aが10の倍数であればBit, そうでなければArrowと表示するプログラムを作りなさい。ただし、Aが50の時に限り、BitArrowと表示しなさい。

それぞれの問題の模範解答と採点基準を示す。

#### 問題(1)の模範解答

```
for(i=1;i<=12;i++){
    addText("j", "<img src=images/neko1.png>");
}
```

問題(1)では反復構造と画像表示を使って、以下を調べることにした。

- (1) Forのブロック構造を作ることができるか
- (2) カウンタ変数の部分を正しく記述できるか
- (3) 画像を表示するという処理を正しく記述できるか

#### 問題(2)の模範解答

```
if(A==50){
    addText("j", "BitArrow");
}else if(A%10==0){
    addText("j", "Bit");
}else{
    addText("j", "Arrow");
}
```

問題(2)では条件分岐とFizzBuzzに類似した問題を使って、以下を調べることにした。

- (1) IFのブロック構造を作ることができるか
- (2) 分岐に関わる処理条件を正しく設定できるか
- (3) メッセージを表示するという処理を正しく記述できるか

調査3では、1時間目から4時間目までの授業の最後に、それぞれアンケートを実施し、学習者が授業内容をどのように感じたのかを難しさと楽しさからそれぞれ4段階で調査する。また、その時間に感じたことなどを自由記述さ

せる。

以上の3つの調査結果を総合し、BitArrowの学習環境としての評価と、プログラミング教育に必要な工夫点を考察する。

## 4. 結果

### 4.1 調査1：授業時の様子

授業時の生徒の様子については、『従来型の学習法の課題』が解決したかと、『DOM利用による学習態度』の2点を観察した。

『従来型の学習法の課題』については、テキストエディタとブラウザを組み合わせていた時のような、ウィンドウ操作とファイル操作による躓きは一切見ることなく、学習環境に入ってしまったら、学習内容に集中させることができた。また、エラーメッセージが表示されるので、生徒は自分でプログラム上のミスを探し、それを修正しようとする生徒の姿勢を観察できた。ただし、一部の生徒が、本来HTMLタブ内に書くべき記述をJavaScriptタブ内に書いてためにエラーを生じると共に、その原因を特定できなくなることがあった。

『DOM利用による学習態度』については、プログラムの実行を一時的に停止させるwaitという命令の効果に注目した。従来のJavaScriptでプログラムの実行を一時的に停止させるためには、setInterval()メソッドやclearInterval()メソッドを用いて複雑な処理を行う必要であったが、BitArrow版JavaScriptではwait(時間)という1つの命令で、簡単に実現する。引数である時間の単位はミリ秒であるため、wait(1000)と記述することで1秒間プログラムが停止する。この命令を繰り返すの間に書き加えるだけで、処理結果のすべてが一度に表示されるのではなく、結果の一つ一つが1秒おきに表示されるようになる。プログラムを作る立場にすると、結果を一つ一つ確認しながら、その正誤に一喜一憂する状況が作られることになる。このwait命令を紹介すると多くの生徒が、他のプログラムの中にも組み込み、時間の設定をし直したり、処理結果の一つ一つを凝視するようになった。例えば、図10はFizzBuzz問題において、1から順にカウントアップし、15で“FizzBuzz”と表示された時の喜びを表している様子である。他にも、両手を合わせて祈ったり、全身で喜びを表現するような生徒も観察された。

### 4.2 調査2：テスト

テスト問題(1)の結果を表2と図11に示す。

繰り返しは再三学習を行ってきた内容であったが、forを使うことや、{}を用いて適切に繰り返しの範囲を定義することができない生徒が多く、正しくブロック構造を作ることができた生徒は33.1%、カウンタ変数としてforの()内に「i=1;i<=12;i++」と正しく記述できた生徒は22.1%に



図10 プログラムの実行結果に満足して喜ぶ生徒

表2 テスト問題(1)の結果 (n=154)

	正答者数	正答率
観点1 『for文と{}の使い方』正答者数/率	51名	33.1%
観点2 『カウンタ変数』正答者数/率	34名	22.1%
観点3 『繰り返される処理』正答者数/率	23名	14.9%
『完全解答』正答者数/率	12名	7.8%

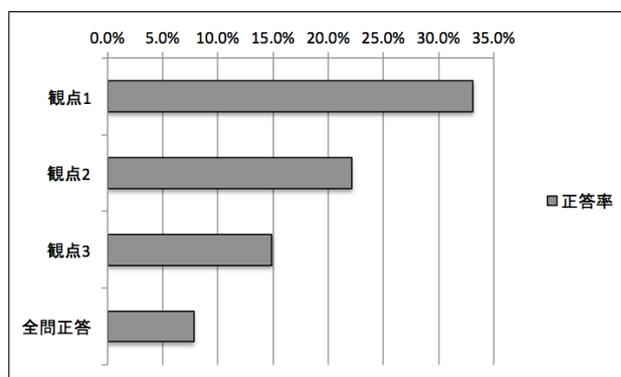


図11 テスト問題(2)の結果

過ぎなかった。また、画像を表示させるための命令を正しく記述できた生徒も14.9%に留まったが、addTextだけを見ると半数以上の生徒ができていた。一方、7.8%の生徒がすべてを正しく記述できた。

採点の例としては、次のようにforを使わず画像表示命令を必要数書いても同様の結果は得られるが、反復構造を実装できなかったと判断して『for文と{}の使い方』と『カウンタ変数』は誤りとした。ただし、『繰り返される処理』の画像表示部分はaddTextとimg srcができていたので正解とした。またaddTextとimg src部分は今回は両方ができて正解としたが、片方だけできているという解答も多く、特に、addText部分のみできている生徒は半数を超えていた。

問題 (1) の誤り例

```
addText("j", "<img src='images/neko1.png'>");
addText("j", "<img src='images/neko1.png'>");
addText("j", "<img src='images/neko1.png'>");
addText("j", "<img src='images/neko1.png'>");
(中略)
addText("j", "<img src='images/neko1.png'>");
```

テスト問題 (2) の結果を表 3 と図 12 に示す。

表 3 テスト問題 (2) の結果 (n=154)

	正答者数	正答率
観点 1 『if 文との使い方』正答者数/率	48 名	31.2%
観点 2 『条件設定』正答者数/率	25 名	16.2%
観点 3 『分岐先の処理』正答者数/率	31 名	20.1%
『完全解答』正答者数/率	12 名	7.8%

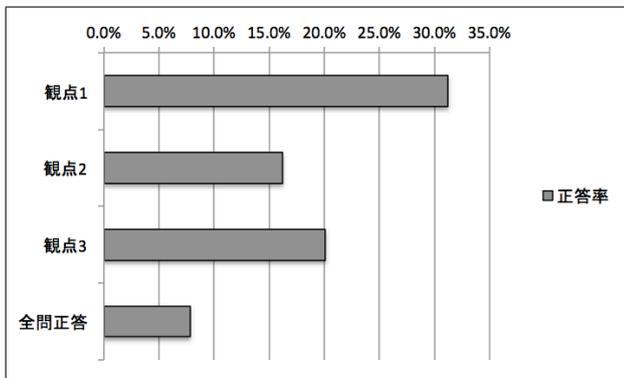


図 12 テスト問題 2 の結果

条件分岐も再三学習を行ってきた内容であったが、if を正しく使うことや、{} を用いて適切に分岐の範囲を定義することができない生徒が多く、正しくブロック構造を作ることができた生徒は 31.2%、分岐させるための条件を正しく設定できた生徒は 16.2% に過ぎなかった。また、分岐先での処理を正しく記述できた生徒も 20.1% に留まったが、7.8% の生徒がすべてを正しく記述できた。誤りの例として次のように if 文の構造を作ることができず独自の表記法を作ってしまうものや、{} の対応関係が適切に使えていないものが多かった。

問題 (2) の誤り例

```
A=rnd(100);
if("j"=10,20,30,40,60,70,80,90)
addText("j"=Bit);
if("j"=50)
addText("j"=BitArrow);
if("j"=1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,
(中略)
91,92,93,94,95,96,97,98,99)
addText("J"=Arrow);
```

4.3 調査 3: 授業アンケート

各授業の終わりに授業の難しさと楽しさについて実施したアンケートの結果を示す。

表 4 アンケート「授業の難しさ」の結果 (n=154)

組	とても難しい	少し難しい	あまり難しくない	簡単
1 時間目	11 名 7.1%	75 名 48.7%	57 名 37.0%	11 名 7.1%
2 時間目	23 名 15.1%	92 名 60.5%	35 名 23.0%	2 名 1.3%
3 時間目	20 名 13.0%	89 名 57.8%	40 名 26.0%	5 名 3.2%
4 時間目	69 名 47.3%	61 名 41.8%	16 名 11.0%	0 名 0.0%

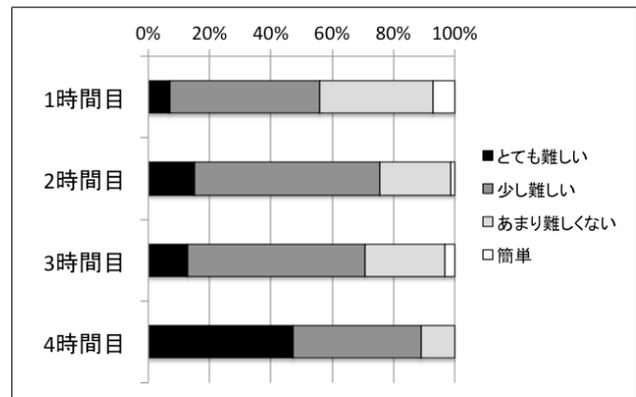


図 13 アンケート「授業は難しかったですか?」の結果

1 時間目から 4 時間目まですべての授業で「とても難しい」と「少し難しい」を合わせて 50% を越えていて、生徒の多くが難しさを感じながら授業を受けていた。その中でも、4 時間目の FizzBuzz を作る時間が最も難しいと感じていた。

表 5 アンケート「授業の楽しさ」の結果 (n=154)

組	とても楽しい	少し楽しい	あまり楽しくない	つまらない
1 時間目	49 名 31.8%	92 名 59.7%	57 名 7.1%	11 名 1.3%
2 時間目	54 名 35.5%	89 名 58.6%	7 名 4.6%	2 名 1.3%
3 時間目	61 名 39.6%	87 名 56.5%	5 名 3.2%	1 名 0.6%
4 時間目	69 名 33.6%	61 名 52.7%	16 名 11.6%	0 名 2.1%

楽しさについては、1 時間目から 4 時間目まですべての授業で「とても楽しい」と「少し楽しい」を合わせて 80% を越えていて、生徒の多くが楽しさを感じながら授業を受けていた。

自由記述では、工夫することや上手くいったことの喜びを示すコメントが多かった。特に授業の楽しさとの関連で、「楽しい」と答えた生徒には「思い通りにいくと達成感があった」や「できたときとてもうれしかった」など、プ

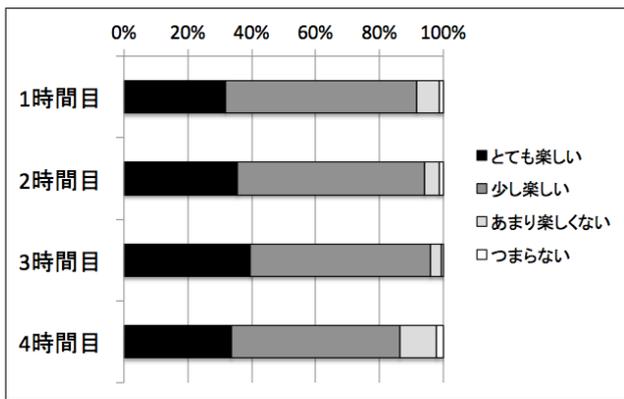


図 14 アンケート「授業は楽しかったですか？」の結果

プログラミングが上手いといったことをその要因にあげる生徒が多かった。逆に「楽しくない」と答えた生徒には「できなかった」「全部エラーになる」「難しかった」とプログラミングが上手く行かなかったことをその要因にあげる生徒が多かった。

## 5. 考察

3通りの調査結果を総合し、BitArrowの有用性及び授業法について評価する。

従来型のテキストエディタとブラウザを組み合わせた学習法では、コンピュータに不慣れな初学者がウィンドウ操作やファイル操作に戸惑ってしまうことが多かったが、BitArrowで行った授業では、開始後すぐに生徒は学習内容に集中することができた。教員とすれば全員が揃わなければ先に進むことができないため、これまでは不具合が生じた生徒の対応に負われることで、授業進度に影響したり、すぐにできてしまう生徒に待ち時間を生じさせたが、そのような問題は大幅に軽減された。

また、wait命令を入れて処理結果を1つずつ表示させるだけで、生徒の活動状況が活気づいた。これは、一瞬で処理結果が表示されるよりも、所謂、ワクワクする感じが生まれたものと考えられる。特に、FizzBuzz問題で「15の時にFizzBuzzと表示されるかどうか」にこの傾向が強かったが、ifの条件設定が適切でなければFizzBuzzと表示されないことから、プログラムを正しく作るという本来の学習目的を果たすためにwait命令が効果を発揮したと言える。

学習内容の理解については、今回のテストでは「すべて自分で作る」ことを目標とした場合には、十分に達成できなかった。特に、for文もif文も{}を用いてブロック構造を適切に作るができない生徒が多かった。その一方で、addTextやimg srcなど個々の命令を正しく記述できる生徒が多く、理解できる事柄に順番や流れがあることが示唆された。今回の授業を実践した報告のように限られた時間で一定レベルの目標を設定する場合には、時間配分や力点の置き場所への配慮が必要ことがわかった。

生徒の多くは、プログラミングの経験を持っていなかった

たが、難しいと感じながらも楽しく授業を受けることができたと感じていた。生徒の自由記述には「仕組みがわかった」「自分でできた」とする記述が多いことから、日常的に利用している情報技術に関連付けながら、達成感を与える授業が実践できた。その背景にはBitArrowの持つ操作のしやすさやDOMの利用、エラー表示による生徒自身の対応力の強化があると考えられる。

本研究で実施した内容は情報科の教科書にも準拠したものである。また、今回扱わなかった配列や関数などを取り入れることで、プログラム作りの可能性も向上し、より高度なプログラムを作ることができるようになる。その導入として実践した内容は効果的であると考えられる。BitArrowの利用と合わせて今後の高等学校におけるプログラミング教育の一手法として提案したい。

## 6. まとめ

オンラインプログラミング学習環境BitArrowの有用性を評価し、従来の課題を解決できることを確認した。今後も社会的な要請の度合いが増すであろうプログラミング学習の基盤として提案したい。また、本研究で行ったような入門授業以外にも、より時間をかけた発展的な授業や、個人作品を作らせる授業、家庭での学習を想定した授業など、様々な授業にもBitArrowは適用できると考えている。今後もその可能性を追求すると共に、効果的な授業方法を模索していきたい。

## 謝辞

本研究は、科学研究費補助金(基盤研究(C)25350214, 奨励研究16H00221)F, 並びに、パナソニック教育助成財団第42回の助成を受けています。

## 参考文献

- [1] 幼稚園, 小学校, 中学校, 高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について(答申)(中教審第197号)平成28年12月21日中央教育審議会:  
[http://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo0/toushin/1380731.htm](http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/1380731.htm)
- [2] 田中健: 文部科学省プログラミング教育実践ガイド・普通科高校でのWebプログラミング  
[http://jouhouka.mext.go.jp/school/programming\\_zirei/](http://jouhouka.mext.go.jp/school/programming_zirei/)
- [3] 間辺広樹, 長島和平, 長慎也, 並木美太郎, 兼宗進: 高等学校における複数言語によるプログラミング教育の提案～情報システムの理解を目標としたドリトル, JavaScript, PHPの連携～研究報告コンピュータと教育(CE), 2016-CE-133(3), pp.1-10 (2016).
- [4] 長島和平, 長慎也, 間辺広樹, 並木美太郎, 兼宗進: JSLesson - 高校生向けJavaScript学習環境, 情報処理学会研究報告コンピュータと教育, 2016-CE-134,16, pp.1-9 (2016).
- [5] 兼宗研究室: プログラミング言語「ドリトル」.  
<http://dolittle.eplang.jp/>