

高等学校における教育用 JavaScript の授業実践報告

—オンラインプログラミング学習環境 Bit Arrow を用いて—

間辺広樹

神奈川県立柏陽高等学校

実践の背景

次期学習指導要領では、小学校から高等学校までのすべての校種でプログラミングが必修化されることになった。2016年12月に出された中央教育審議会の答申では「自分が意図する一連の活動を実現する」ことを**プログラミング的思考**とし、それを育む教育を求めた。高等学校では、現在は情報科の選択科目である「情報の科学」においてのみプログラミングを学習することとなっているが、この科目を開講している高等学校は2割程度にとどまっている。次期学習指導要領では新設される共通必修科目「情報I」にプログラミングが位置付けられ、すべての高校生がプログラミングを学ぶこととなった。「自分が意図する一連の活動を実現する」ことは「自分でプログラミングできるようになる」と解釈できるが、この目標は高い。本誌にて何度も取り上げられているように、基礎とはいえ、次のような概念を扱い、少ない授業時間数で「できるようになる」ことを高校生に求めるのは簡単ではない。

- 変数、配列
- 制御構造(順次、条件分岐、反復)
- アルゴリズム(合計、探索、整列など)

特に、近年はスマートフォンの普及などに伴って、高校生のコンピュータに対する操作能力は著しく低下していることも円滑に授業を進められない要因となっている。この傾向は今後もより顕著になると考えられる。

その一方で、「プログラミングができるようになりたい」という生徒の声も増えている。アンケー

トでも「やったことはないけれども、できるようになりたい」というコメントを多く見るようになった。学習目標は高い方が良いが、まずは「プログラミング嫌いを作らないこと」や「自分にもできると感じさせること」が大切ではないかと考える。

ではどのように授業を行っていけばいいのであろうか？ 特に、プログラミングの学習には**言語・学習環境・授業内容**が生徒の学習に大きく影響する。

本稿では言語として JavaScript を扱っている。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理由である。ただし、従来の JavaScript の学習は、操作やエラー表示の点などに課題があった。そこで、それらを解決したオンラインプログラミング学習環境 Bit Arrow を用いることにした。制御構造の基本を理解させることを目標とした4回の授業例を紹介する。

JavaScript 学習の課題

JavaScript は、テキストエディタと Web ブラウザがあれば実習できることから、複数の教科書で使われているが、プログラムの作成は次のようにファイル操作やウィンドウ操作が必要になる。そのため、操作の不慣れな生徒は戸惑うことが多かった。

1. エディタでプログラムを入力する
2. 拡張子を html にして保存する。
3. ブラウザに読み込み、動作を確認する。
4. 修正・更新は「プログラムを書き換え、上書き保

存し、ブラウザでリロードする」を繰り返す。

また、プログラムにミスがあってもエラーが表示されず、その特定も困難であった。

筆者はこれまでの授業実践から、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。

オンライン学習環境 Bit Arrow

□ Bit Arrow の特徴

Bit Arrow¹⁾ は、東京農工大学、明星大学、大阪電気通信大学によって共同開発されたオンラインプログラミング学習環境である。現在、教育用 JavaScript, C, ドリトルの 3 言語から選択して利用することができる。インターネット上のサーバで動作するため、次のようなメリットがある。

- 教師が生徒の利用状況を把握できる
- 複数の言語を利用できる
- 授業の時間外でも学習ができる

□ 教育用 JavaScript

Bit Arrow で使える教育用 JavaScript には従来の JavaScript の問題を解決するために、次の機能が備えられている。

1. HTML と JavaScript のタブを分けている
 2. エラーとその発生個所を表示する
 3. 入力補助機能がある
 4. 自動インデント機能がある
 5. QR コードが生成され、生徒が所有するスマートフォンから実行結果を確認することができる
- 特に、エラーの表示があるため、誤り個所を特定

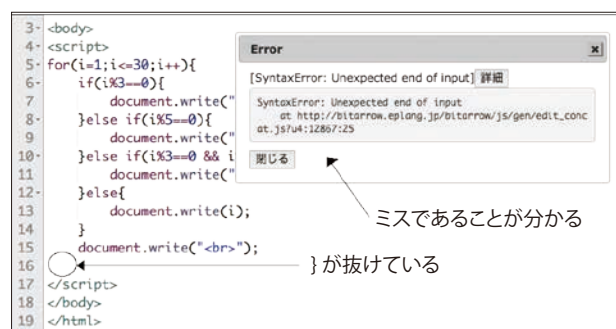


図-1 Bit Arrow 版教育用 JavaScript

がしやすくなった(図-1)。

さらに、通常の JavaScript では多くの文字を入力しなければ実現しない処理が、簡略化されている(表-1)。これを用いることで、プログラムにアニメーション効果をつけることなどが容易になっている。

授業内容

□ 授業計画と工夫

4 クラス 154 名を対象に、65 分の授業を週に 1 コマずつの合計 4 回行う授業を表-2 のように実施した。各授業の工夫点や生徒の様子を説明する。

□ 1 時間目(順次の理解)

1 時間目は、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さまざまな処理ができることを理解させることを目標とした。生徒の多くはプログラミング経験がないため、導入として日本語で記述できるドリトルを利用した。

前半は「歩く」や「右(左)回り」という命令を順に並べることで、図形を描画できることを体験させ、オリジナル作品を作る時間を設けた。後半は授業支援ツールを使い、生徒作品を 10 秒ずつスクリーンに表示して、相互に閲覧し、単純な命令でも組合せ

	従来	教育用
乱数	<code>Math.floor(Math.random() * 100);</code>	<code>rnd(100);</code>
一時停止	<pre>countup = function(){ console.log(count++); } setInterval(countup,100);</pre>	<code>wait(100);</code>

表-1 従来と教育用 JavaScript の比較

時間	言語	目標	主な学習活動
1	ドリトル	順次の理解	描画プログラム制作 生徒作品の相互閲覧
2	JavaScript	分岐の理解	HTML 実習 (Web ページの仕組み) じゃんけんプログラム制作
3	JavaScript	反復の理解	画像表示プログラム制作 FizzBuzz を声に出す体験
4	JavaScript	問題演習	2・3 回目の復習 FizzBuzz プログラム制作

表-2 65 分×4 回の授業





図-2 じゃんけんプログラム

次第でさまざまなものが作れることや、アイデアを出すことが大切であることを説明した。

□ 2 時間目(分岐の理解)

2 時間目は、JavaScript を身近に感じさせることと、条件分岐の構造を理解させることを目標とした。そのために、Bit Arrow の教育用 JavaScript を用いた。

前半は body タグの属性を使い、ページの背景色を 16 進表記で設定させた。そして、Web ページは文字で作られたものが Web ブラウザで組み立てられて我々が目にするものになっていることを説明した。その上で、JavaScript が HTML に埋め込まれて使われている身近な言語であることを、いくつかの Web ページのソースを見せて説明した。

後半は教育用 JavaScript の導入として乱数が関数 rnd によって使えることと、剰余が演算子 % から求められることを説明した。また、乱数で作った整数が奇数か偶数かを判断させるプログラムを通して、条件分岐の構造を説明した。さらに、これを応用してじゃんけんプログラムを作ってみるように指示した(図-2)。

□ 3 時間目(反復の理解)

3 時間目は、反復構造で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。前半は同一の画像を複数個表示させる題材を扱った(図-3)。最初に画像表示の方法と、同じプログラムをコピーすることで、複数個の画像が表示されることを示した。その上で、反復構造を用いることで効率よくプログラ



図-3 画像表示プログラムの例

ミングできることを示した。その際に wait 命令を説明すると、生徒の多くが時間間隔を調整して、画像のアニメーション表示を楽しんだ。

後半は FizzBuzz を扱う 4 時間目の授業のために、生徒を 2～4 人程度のグループに分けた。「3 で割り切れたら Fizz, 5 で割り切れたら Buzz, 3 でも 5 でも割り切れるときは FizzBuzz」というルールの説明をした後、1 から順に声に出して確認するグループ活動をさせた。

最後に、どのようにすれば FizzBuzz プログラムが作れるかを考えさせた。

□ 4 時間目(問題演習)

4 時間目は、分岐と反復の組合せでさまざまなプログラムが作れるようになることを理解させることを目標とした。前半は 2 回目と 3 回目で扱ったプログラムを改めて作らせて、分岐と反復を復習させた。

後半は、3 時間目の FizzBuzz がこれまでの学習内容を組み合わせれば作れることを伝え、自力で作るよう指示した。その際、スクリーンに以下のヒントを少しずつ書き加えた。

- (1) 1 から 30 までを順に表示する
- (2) 3 の倍数だけのときだけ Fizz と表示する
- (3) 5 の倍数だけのときだけ Buzz と表示する
- (4) 3 と 5 の倍数のときに FizzBuzz と表示する

この課題は (3) までは簡単にできても (4) が難しい。ソースコードで一番上に書かなければ正しく動作しない FizzBuzz を表示する処理を、多くの生徒が一番下に書いてしまうからである(図-4)。その際、多くの生徒があえて使う必要のない wait 命令を使い、数や文字を 1 つずつ表示させていた。「15 のときに FizzBuzz と表示されるか否か」で、祈るように手をあわせる生徒や、うまくできたことに大

HTML
JavaScript
FizzBuzz別ページで表示

```

1- for(i=1;i<=30;i++){
2-     if(i%3==0){
3-         addText("hky","Fizz");
4-     }else if(i%5==0){
5-         addText("hky","Buzz");
6-     }else if(i%3==0 && i%5==0){
7-         addText("hky","FizzBuzz");
8-     }else{
9-         addText("hky",i);
10-     }
11-     addText("hky","<br>");
12-     wait(200);
13- }
14- addText("j","---Owari---");

```

実行画面ダイアログ

Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
Fizz ←
16
17
Fizz
19

図-4 誤った FizzBuzz プログラムの例

喜びする生徒もいた。

□ アンケート調査

毎回、授業の楽しさと難しさを4段階で自己評価させた。集計結果を図-5と図-6に示す。

この結果から、生徒は難しさを感じながらも楽しく授業を受けていたことが分かった。特に FizzBuzz を作った4時間目が最も難しさを感じていた。また、授業中の発話から複数の生徒が自宅学習を進めていたことも分かった。

実践のまとめ

オンラインプログラミング学習環境 Bit Arrow を使った4回の授業を紹介した。紙面の関係で、生徒の主観的評価しか載せていないが、筆記テストで取得した客観データでも Bit Arrow の有用性を確認できた。

今回は高校生のプログラミング教育のきっかけとなる入門授業を紹介した。ブラウザで実習できるオンラインプログラミング環境は、興味を持った生徒

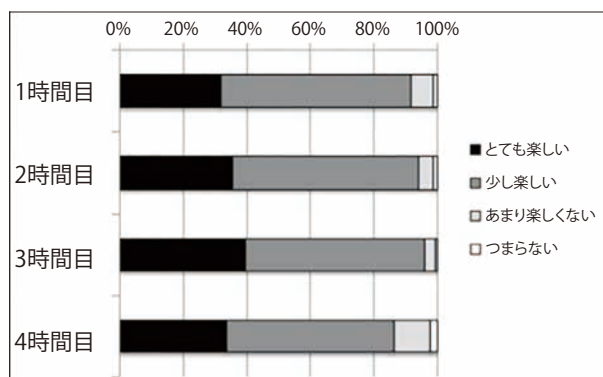


図-5 授業の楽しさ (N=154)

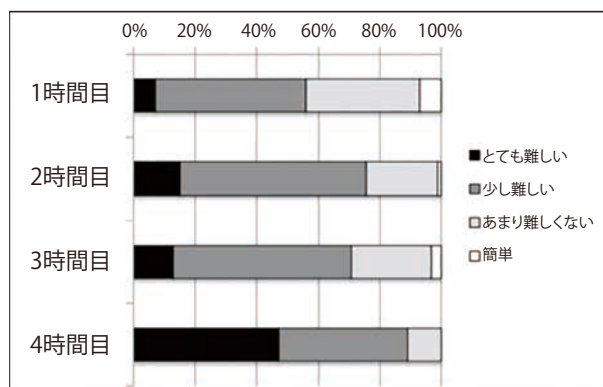


図-6 授業の難しさ (N=154)

が自宅などで学習を進められるなどのメリットが大きいことを確認できた。今後もさまざまなプログラミング教育の手法を模索していきたい。

参考文献

- 1) 長島和平, 本多佑希, 長 慎也, 間辺広樹, 兼宗 進, 並木美太郎: オンラインで複数言語を扱うことができるプログラミング授業支援環境, 情報教育シンポジウム (SSS2016), pp.137-140 (2016).

(2017年8月17日受付)

間辺広樹 (正会員) manaty2005@mh.scn-net.ne.jp

大阪電気通信大学医療福祉工学研究科博士後期課程修了。博士(工学)。神奈川県立柏陽高等学校数学科・情報科教諭。

