

# JavaScript を用いた中学生向けプログラミング教育

山 守 一 徳\*

Programming Education for Junior High School Students using JavaScript

Kazunori YAMAMORI

## 要 旨

日本では、小学校からプログラミング的思考を取り入れた授業を 2020 年までに展開するように要請されているが、中学校ではプログラミング教育をどのように実施したら良いか悩ましい問題がある。小学校向けには Scratch を利用したプログラミング教育が多く行われ始めており、筆者は、その題材を提供してきているが、中学生向きには Scratch によるプログラミングでなく、もう少し進んだプログラミングを教えた方が良いと考える。そこで、本論文では、中学生向けの JavaScript プログラミングを提案する。

## 1. はじめに

プログラミング教育は、今後の日本の技術レベルを上げるには必須のものであろうと気が付き始め、小学校の段階から授業の中に取り入れるように指示が出ている。現状では、プログラミング言語の授業は、大学入学後によく履修する場合がほとんどであり、学生によっては理解できずにプログラミングを毛嫌いする学生も多い。その中で、小学校段階からプログラミング的思考を育む授業を実施するように指示が出ても、学校現場では戸惑いが多いのが実情である。筆者は、これまで小学校向けに Scratch を使った授業実践を行ってきており、小学校向けには多くの素材を提供してきているが、中学生向けには Scratch プログラミングよりも一歩進んだ素材を用いた方が良い。Scratch では、大学の授業で使われているプログラミング言語とは程遠く、Scratch は小学生向けのツールであるからである。そこで、中学生向けに JavaScript を用いたプログラミング教育を行うことを提案する。JavaScript は、Web ブラウザで動作確認することができ、中学校に通常存在しているパソコン上で特別なソフトをインストールする必要もなく実践できるからである。本文では、実際に利用する JavaScript プログラムの具体策を提示し、授業実践の進め方について述べる。

## 2. JavaScript プログラミング

JavaScript は、HTML ファイルの中に直接書き込むか、別ファイルに JavaScript 言語を記載しそのファイルを HTML ファイルの中へ呼び込む形にすることで、プログラミングを行う。テキストエディタ（メモ帳、K2Editor 等）を使って、プログラム文字列を打ち込み、Web ブラウザ（Internet Explorer、Firefox、Chrome 等）を使って、動作させる。WWW サーバ上に HTML ファイルや JavaScript ファイルを置いて、クライアント端末上の Web ブラウザからアクセスする形態にする必要はなく、生徒の端末上に HTML ファイルや JavaScript ファイルを直接置いて、その端末上の Web ブラウザからアクセスするだけで良い。

---

\* 三重大大学教育学部情報教育講座

中学校の端末教室の個々の端末だけで十分に利用することができる。

デバッグの方法は、用いる Webb ブラウザによって少し異なるが、Internet Explorer の場合は、右上の歯車ツールの中の「開発者ツール」にチェックを入れる。Firefox の場合は、右上のメニューの中の「開発ツール」を選び、「開発ツールを表示」をクリックする。Chrome の場合は、右上の「Google Chrome の設定」を選び、「その他ツール」の中の「デベロッパーツール」をクリックするが、使い勝手はあまり良くない。

JavaScript 言語は、Java 言語に似たところがあり、逐次処理、繰り返し処理、条件分岐処理などのプログラムの流れを理解するだけでなく、変数、配列、関数定義、関数呼び出しなどのプログラムの書き方を理解する必要があるため、将来本格的にプログラミングを行う時に知識が大いに役に立つ言語である。

### 3. 他のツールとの比較

小学校向けには Scratch が流行っているが、それ以外にプログラミング教育用として考えられるツールを紹介する。いずれもプログラミングを小学生に対して教える場面で使うことを目指しており、中学生向けには物足りない感じがする。

#### (1)Code Monkey

猿がバナナを取りに行く動きをするゲームの感覚でプログラミングを学習する。イスラエルで開発された商品として安価に売られているが、日本語に対応したオンライン体験版が提供されている。プログラムのコードを書くという操作をさせるので、プログラミングに近い。

#### (2)Lightbot

ロボットが電灯を点灯しに行く動きをするゲームの感覚でプログラミングを学習する。マウスでブロックを組み立てていくという操作をするので、低学年向きである。コードを書く必要がなくプログラムのロジックを学習するのに有効である。最初は、Flash のゲームとして提供され、その後 iOS と Android 向け無料ゲームとしても提供されている。

#### (3)Osmo

iPad か iPhone の前で、ブロックを並べると、iOS のカメラがそのブロックを読み取って動作するもので、音楽を作成する Coding Jam やキャラクターに苺摘みの冒険に行かせる動きをする Coding Awbie というソフト等がある。専用ブロックを使うので費用が掛かり、低学年向きである。

#### (4)Robot Turtles

パソコンを使わないカードゲームであり、亀に宝石を取りに行かせるために「前へ進む」「右へ回る」等のカードをボード上に置かせるゲームである。親と子どもで遊ぶような使い方をし、子どもが提示したカードに従って亀を移動させる操作を親が行う。カードを選び並べるという操作がプログラミングに近い。

#### (5)Viscuit

Scratch に似ているが、ブロックを操作して動作させるのではなく、絵を描いて、眼鏡のイラストの中に 2 つの絵を貼り、左右の 2 つの絵の違いをコンピュータが動きに変えて見せてくれる。眼鏡を複数個使って複雑な動きにすることができる。

### 4. 提案する授業実践サンプル

「じゃんけんゲームを作ろう」のタイトルで、JavaScript プログラミング体験を行うことを提案する。

短いプログラムで動くことが必要であり、興味を持たせて意欲的に取り組ませるには最適と考える。

#### 4. 1 第1段階

最初の段階は、Hello World! の文字列を表示させるプログラムをテキストエディタ（メモ帳、K2Editor 等）を使って、打ち込ませる。図1にそのプログラムを示す。ファイル名は start.html といった名前でも html ファイル形式で良い。

図1の中の3行目の氏名は、生徒の名前、9行目の〇年〇組は、所属するクラスを入力させる。

このファイルを Web ブラウザで表示させる。すなわち、start.html が、d ドライブの直下に存在するならば、file:///D:/start.html と Web ブラウザの URL 欄に入れるだけで良い。

図2に、「開発ツールを表示」状態にした場合の実行結果の画面を示す。氏名と Hello World! と〇年〇組の3行が表示された状態になる。この3行の部分は、図1の中の3行目の氏名と6行目の Hello World! と9行目の〇年〇組に対応している。重要なのは、6行目の document.write ("Hello World!"); の実行によって、Hello World! が画面に出力されることであり、document.write という関数が画面出力させる機能を持つことである。この第1段階では、関数が事前に用意されていて、それ呼び出して使うにはどうやって書くのかを教える。JavaScript のプログラムは、5行目の<script type="text/javascript">と7行目の</script>で囲まれた部分に書くことによって実行される。この6行目の部分を、以後、拡大していけば複雑なプログラムを書くことができる。JavaScript の部分は、動作後には HTML による記述に置き換えられ、その HTML の文が Web ブラウザで表示される。そのことを教えれば、図1のプログラムは、図3の HTML ファイルを書いていることと同じであると理解できる。HTML の文章を教える必要があるが、<html><body></body></html>は定型の決まり文として教え、<br>と書くことと改行されることを教えるだけで済む。

#### 4. 2 第2段階

次の段階では、じゃんけんゲームに近付けるために、「グー」「チョキ」「パー」のいずれかが表示されるようにする。乱数の発生関数を教え、乱数の出た値によって、条件分岐させて「グー」「チョキ」「パー」を表示させる。

```
<html>
<body>
氏名
<br>
<script type="text/javascript">
document.write("Hello World!");
</script>
<br>
〇年〇組
</body>
</html>
```

図1 初期プログラム

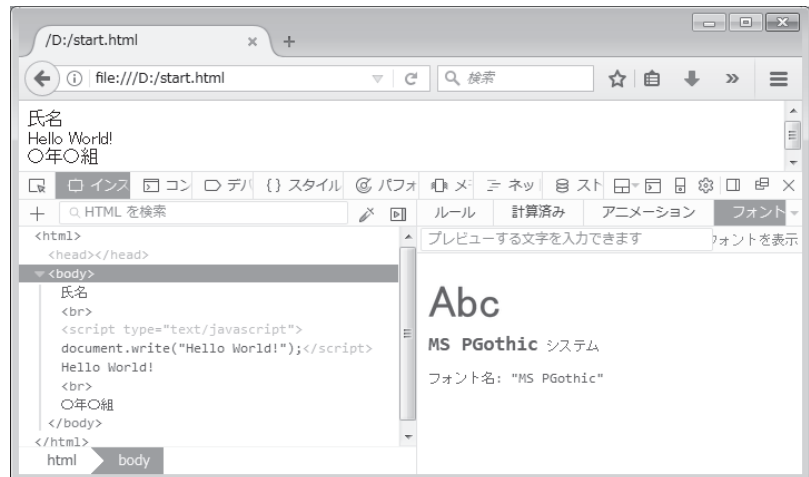


図2 初期画面（Firefox の場合）

```
<html>
<body>
氏名
<br>
Hello World!
<br>
〇年〇組
</body>
</html>
```

図3 動作後の HTML

図4に第2段階のプログラムを示す。初期のプログラムの `document.write ("Hello World!");` の行の下に追加させる。ここで重要なのは、if文による分岐と `Math.random()`関数と `Math.floor()` 関数の動きである。`Math.random()`と書くことによって、関数が呼び出され0以上1未満の小数点の値が返されてくる。`*3`で3倍することで0以上3未満の小数点の値になり、`Math.floor()`関数に渡すことで、小数点以下切り捨ての値が返されてくる。`var ransuu` で `ransuu` という名前の変数を用意し、そこに0か1か2の値が代入される。if文の書き方を教えて、0か1か2の値によってグーかチョキかパーが出力される。if文の条件分岐は、プログラミングを教える時の基本であるので、ここまででプログラミングというものはどういう作業をするのかを把握できて、今後にも繋がる能力を身に付けたことになる。図5に、「開発者ツール」を表示状態にした場合の第2段階の実行結果の画面を示す。`Hello World!`と〇年〇組の間に、「グー」「チョキ」「パー」のいずれかが表示される。7行目に `document.write("<br>");` を入れることによって、`Hello World!`の行の下にじゃんけんの手が表示され、`document.write("<br>");` を入れないと、`Hello World!` の右にじゃんけんの手が表示されるので、`document.write("<br>");` の役目も理解できると思われる。

#### 4. 3 第3段階

第3段階では、配列の使い方を教え、第4段階へ繋げたい。第2段階と同じ画面出力になるが、いろいろな書き方があることも知ることになる。図6に第3段階のプログラムを示す。プログラムの行数では、第3段階の方が第2段階よりも短くなる。`janken` という変数を用意し、1番目の要素に"グー"を2番目の要素に"チョキ"を3番目の要素に"パー"を入れた配列を代入しておけば、`janken[ransuu]`によって、要素の値を取り出すことができる。この時注意が必要なのは、配列の要素を示す

```
<html>
<body>
氏名
<br>
<script type="text/javascript">
document.write("Hello World!");
document.write("<br>");
var ransuu = Math.floor(Math.random() * 3 );
if(ransuu == 0) {
    document.write("グー");
}
else if(ransuu == 1) {
    document.write("チョキ");
}
else {
    document.write("パー");
}
</script>
<br>
〇年〇組
</body>
</html>
```

図4 第2段階プログラム

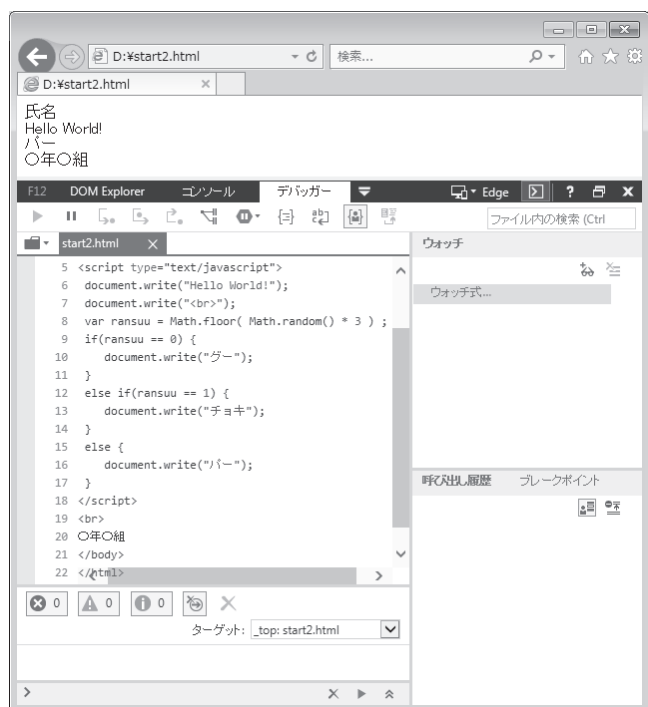


図5 第2段階の画面（IEの場合）

時に 0 から数えることである。先頭の 1 番目の要素は `janken[0]` で取り出される。`ransuu` の値は、0、1、2 のいずれかであるので、1 番目、2 番目、3 番目がそれぞれ取り出される。

#### 4. 4 第 4 段階

第 4 段階では、自分で関数を定義し、その関数を呼び出すことを教え、第 5 段階へ繋げたい。画面上にボタンを配置し、そのボタンを押すことで関数が呼び出されるプログラムとする。図 7 に第 4 段階のプログラムを示す。第 3 段階で配列を知ることができたので、第 4 段階では配列を活用する。関数の定義は、`<head></head>` の中に書くことを教える。`<body></body>` の中は Web ページの本体部分を書く箇所であるが、`<head></head>` の中は本体で使用する準備部分を書く箇所である。`<input` の行は、ボタンを表示させるための HTML 言語の記述文であり、ボタンを押した時に `jankenpon()` 関数を呼び出すための記述文である。

第 4 段階のプログラムを Web ブラウザで表示させた結果を図 8 に示す。ジャンケンポンのボタンが、Hello World! と〇年〇組の間の行に表示され、このボタンを押すと、`document.write ( janken[ransuu] );` の実行によって、グー、チョキ、パーのいずれかの文字のみが表示された画面となる。第 2 段階、第 3 段階では、ページを再表示させるごとにじゃんけんの手が変わるが、第 4 段階では、ボタンを押すことによってじゃんけんの手が変わる。



図 8 第 4 段階の画面

```
<html>
<body>
氏名
<br>
<script type="text/javascript">
document.write("Hello World!");
document.write("<br>");
var ransuu = Math.floor(Math.random() * 3 );
var janken = ["グー","チョキ","パー"];
document.write( janken[ransuu] );
</script>
<br>
〇年〇組
</body>
</html>
```

図 6 第 3 段階プログラム

```
<html>
<head>
<script type="text/javascript">
function jankenpon () {
    var ransuu=Math.floor(Math.random() * 3 );
    var janken = ["グー","チョキ","パー"];
    document.write( janken[ransuu] );
}
</script>
</head>
<body>
氏名
<br>
<script type="text/javascript">
document.write("Hello World!");
</script>
<br>
<input onclick="jankenpon();" type="button" value="
ジャンケンポン" />
<br>
〇年〇組
</body>
</html>
```

図 7 第 4 段階プログラム

#### 4. 5 第5段階

第4段階まで進むと、あとはゲームに近付けるようにしたい。乱数発生で計算機がじゃんけんの手を出し、人間が出すじゃんけんの手と勝ち負けを競いたい。グー、チョキ、パーの3つのボタンを用意し、人間が出す手は、どのボタンが押されたかで判断する。勝ち負けの判定もプログラムで行い、何勝何敗何引き分けかを表示させるところまで行う。図9に第5段階のプログラムを示す。document.body.innerHTML += "&nbsp;" ; の書き方が新しいが、+= の演算子は、document.body.innerHTML = document.body.innerHTML + "&nbsp;" ; と等価であり、" &nbsp;" を後ろに加える形で代入することになる。" &nbsp;" は、空白文字を意味する。document.body.innerHTML は、Web ページの本体の文字列を指し、文字列に対して+の演算子は、文字列を繋げる演算を行う。aikosuu += 1; の場合は、aikosuu が初期値 0 からカウントアップしていく整数の役目を果たしている。document.getElementById("aiko").innerHTML=aikosuu; は、<span id="aiko">0</span> と書かれている部分を aikosuu の値に置き換える動きをなす。

第5段階のプログラムを Web ブラウザで表示させた結果を図10に示す。グー、チョキ、パーの3つのボタンが並び、ボタンを押す度に、1行追加され、「計算機の手」と「あなたの手」と勝ち負けが表示される。何勝何

```
<html>
<head>
<script type="text/javascript">
function jankenpon (dasute) {
    var ransuu = Math.floor( Math.random() * 3 );
    var janken = ["グー","チョキ","パー"];
    document.body.innerHTML += "計算機の手は";
    document.body.innerHTML += janken[ransuu];
    document.body.innerHTML += "&nbsp;";
    document.body.innerHTML += "あなたの手は";
    document.body.innerHTML += janken[dasute];
    document.body.innerHTML += "&nbsp;";
    if(ransuu == dasute) {
        document.body.innerHTML += "あいこ";
        aikosuu += 1;
    }
    else if(dasute - ransuu == 1 || dasute - ransuu == -2) {
        document.body.innerHTML += "あなたの負け";
        makesuu += 1;
    }
    else {
        document.body.innerHTML += "あなたの勝ち";
        kachisuu += 1;
    }
    document.body.innerHTML += "<br>"
    document.getElementById("kachi").innerHTML=kachisuu;
    document.getElementById("make").innerHTML=makesuu;
    document.getElementById("aiko").innerHTML=aikosuu;
}
</script>
</head>
<body>
氏名
<br>
<script type="text/javascript">
    document.body.innerHTML += "Hello World!";
    var kachisuu=0;
    var makesuu=0;
    var aikosuu=0;
</script>
```



敗何引き分けかを示している行  
の中の数字も順次変更されてい  
く。ここまで到達できればゲー  
ムとして完成である。

## 5. 授業進行

第5段階のプログラムは、第4段階のプログラムと比べると行数が多くなるため、打ち込みに時間が掛かり、授業中に全員を成功させることは難しいと思われる。この第5段階は、目標としてプログラムを提示し、進捗の早い生徒だけが取り組む課題とすると良い。第4段階まで到達できるかについても、授業の回数による。第1段階のプログラムは、ソースを見せながら打ち込ませると良いが、第4段階のプログラムまですべて打ち込ませては1回の授業では無理である。2回以上は必要と思われる。1回の授業にするには、第2段階と第3段階のプログラムはソースファイルを渡して、第4段階のプログラムは、第3段階のソースファイルから直す形とするのが良い。第4段階でグー、チョキ、パーのじゃんけんの手が表示されれば、プログラミングの達成感が得られる。生徒全員がそこまで達成できれば、授業として成功であると思う。プログラミングの授業の難しいところは、生徒によって進度が違いすぎるところにあり、早い生徒用の課題を用意しつつ進めないといけない。第5段階のプログラムはそのために活用して欲しい。

```
<br>
<input onclick="jankenpon(0);" type="button" value="グー" />
<input onclick="jankenpon(1);" type="button" value="チョキ" />
<input onclick="jankenpon(2);" type="button" value="パー" />
<br>
○年○組
<br>
<span id="kachi">0</span> 勝 <span id="make">0</span> 敗
<span id="aiko">0</span> 引き分け
<br>
</body>
</html>
```

図9 第5段階プログラム

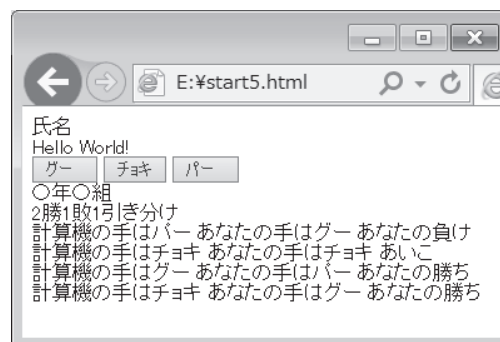


図10 第5段階の画面

## 6. まとめ

中学生向けのプログラミングは、教えるレベルが簡単すぎても役に立たないし、難しすぎても全員が完成しないことになる。先生1名で、クラス全員の生徒に対して、プログラミングを教えると、動かない生徒のプログラムの原因を探るのに時間が取られ、先生が予定通りに授業を進められないことが起きる。端末教室は通常は鍵で閉じられており、生徒が自由な時間に端末に触れる訳でない。宿題として残しても自宅で演習できる生徒と自宅では演習できない生徒に分かれ、宿題の遂行率も低いものとなる。その中で、プログラミング能力を高めるような授業を実践するように文部科学省から指導されても、ノートPCを全生徒に支給する訳でもなく、達成方法は困難を極める。それを補うには、教える内容にアイデアが必要である。今回、プログラムを提示し、それらの参考になれば幸いである。

なお、HTMLもHTML5形式になって仕様が変わってきており、提示したプログラムは、HTML5形式になっておらず、HTMLを教えるという点で利用されると適切でないことに注意して下さい。

## 参考文献

- 1) 狩野祐東：「確かな力が身につく JavaScript「超」入門」SB クリエイティブ（株）（2015）
- 2) 山守一徳：「Scratch を用いた小学生向けプログラミング」高田短期大学キャリア研究センター紀要・年報 3 号 pp.63-72 (2017.3)
- 3) Code Monkey： <https://codemonkey.jp/> (2017.10 現在)
- 4) Lightbot： <http://lightbot.com/> (2017.10 現在)
- 5) Osmo： <https://www.playosmo.com/ja/> (2017.10 現在)
- 6) Robot Turtles： <http://www.robotturtles.com/> (2017.10 現在)
- 7) Viscuit： <http://www.viscuit.com/> (2017.10 現在)