

Javascript を活用した教材作成

山口大学・教育学部 北本 卓也

Takuya Kitamoto, Faculty of Education, Yamaguchi University

1 はじめに

近年、学校現場での ICT の導入が進んできているが、その状況はハードウェアに偏っており、それを活用するためのソフトウェアが不足している。特に授業において ICT を活用するための教材が不足しており、せっかく導入した種々の IT 機器の機能を活かしていない状況にある。最も手っ取り早く教材を増やしていく方法は販売されているものを購入することであるが、教材の中身が固定化されることや予算上の問題などがあり、現実的な解決策とはなりにくい。

現場の教員が自分で簡単に教材が作成することができれば、このような問題の一助となるが、実際には ICT を活用した教材作成は ICT に関わる技術的な知識を必要としており、現場の教員にとって容易でない。そこで本稿では現場の教員が簡単に教材作成が行え、再利用可能であるようなシステム構築について述べる。

具体的には、Javascript を活用することにより Web ブラウザ上での教材作成とするシステムを提案する。

2 Javascript を活用した教材作成システムの特徴

2.1 従来の教材作成システムとの違い

従来使われている E-Learning システムのほとんどは図 1 のように Web サーバーに E-Learning システムや数式処理システムを搭載し、教材作成者と教材利用者がともにその Web サーバーにアクセスして使う形を取っている。

この形の教材作成システムは実用的ではあるが、以下のような問題点を持っている。

- Web サーバーに E-Learning システムや数式処理システムをインストールする必要があるため、そのサーバーの管理者権限が必要である。これは安価なレンタルサーバーは使えないことを意味している。また管理の手間も必要となる。
- 計算や処理が Web サーバー上で行われるため、教員や生徒がシステムを使用しているときにはネットワークへのアクセスが必要である。

今回提案する Javascript を活用するシステムの構成図は図 2 のようになる。

図からわかるようにこのシステムは基本的にパソコンまたはスマホのブラウザ上で動作する。このため、従来の E-Learning システムに比べ、以下のような特徴を持っている。

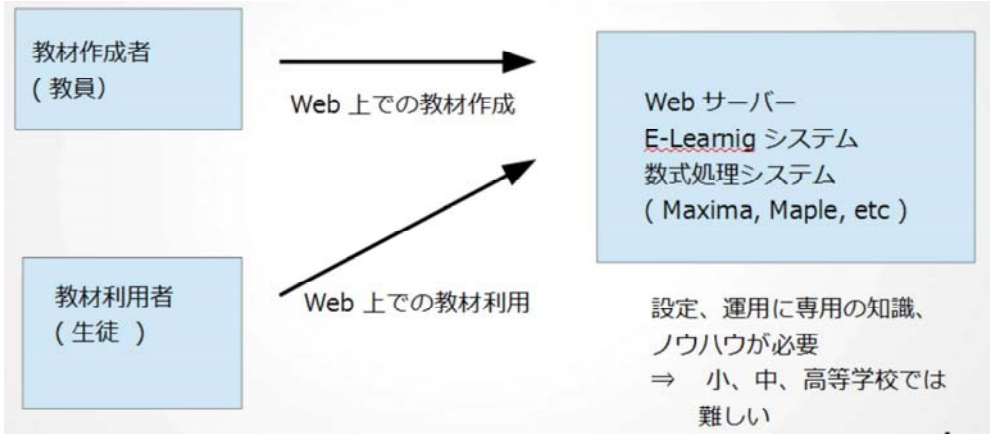


図 1 : 従来のシステムの構成

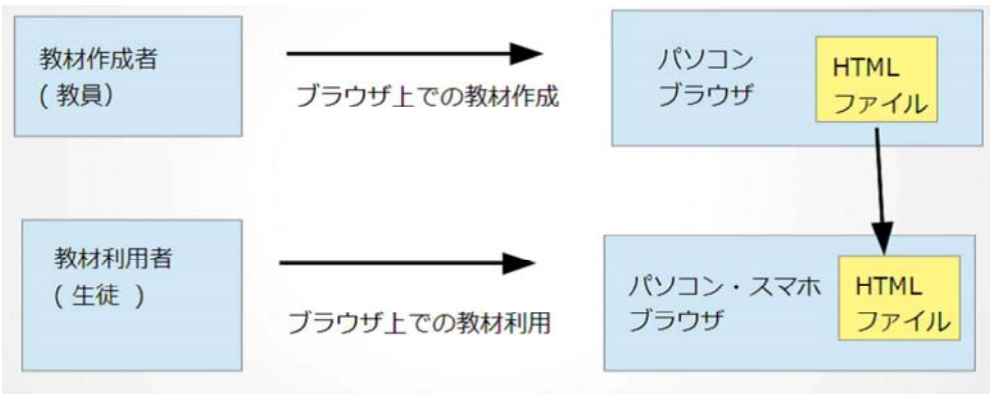


図 2 : 提案するシステムの構成

- 通常の Web サーバーで十分であり、安価なレンタルサーバーを使うことができる。
- 計算や処理がパソコンまたはスマホ上で行われるため、教員や生徒がシステムを使用しているときにネットワークへのアクセスが必要ない。

すなわち、本稿のシステムは下記のような長所を持っている。

- 使用するのにサーバー管理などの専門知識が不要なので、小・中・高等学校でも容易に管理ができる。
- 教室でもネットワーク接続が不要なので、計算機環境を問わない。
- ブラウザ上で簡単に教材作成が可能である。

従来の E-Learning では業者が作成したものをそのまま使用するケースが多かったが、本稿で提案するシステムを用いれば、現場の教員が自分で教材を作成する、もしくは既にあるものを自分の授業に合わせてカスタマイズできるようになると思われる。

2.2 システムの構成

本稿のシステムは基本的には html ファイルと Javascript のファイルからなる。構成は以下のようにになっている。

- プログラム部分は Javascript で記述されている。
- エディタの部分は Quill (Javascript のライブラリ) を用いている。また、数式表示は KaTeX (Javascript のライブラリ) を用いて行われる。
- 数式処理の計算は Algebrite (Javascript のライブラリ) を用いて行われる。
- 暗号化の処理は sha1 (Javascript のライブラリ) を用いて行われる。
- データの保存はブラウザの Web Storage の機能を用いて行われる。

上記からわかるようにシステムは html ファイルと Javascript のライブラリのみを用いて構成されており、Google Chrome, Firefox や Safari などのモダンなブラウザが動作するパソコン・スマホ・タブレットなどの環境であれば動作する。

2.3 システムの使い方

本稿のシステムではブラウザ上で教材を作成する。Javascript の命令を使うことができるので、いろいろな処理 (例えば、乱数の発生) を行うことが可能である。また、Algebrite を取り込んでいるので、数式処理計算が可能で数学の問題の採点もできるため、簡易的な E-Learning システムも作成可能である。具体的な使い方などの詳細は紙面の都合のため本稿では省略するので、参考文献 [5] を見ていただきたい。

3 Cinderella と Javascript との連携

Cinderella では作成した図を html に変換することが可能である。html に変換された Cinderella の図は Javascript を用いて描画されており、ブラウザ上でその図を動かすこともできる。また、CindyScript というプログラミング言語で操作することが可能である。

本稿で提案するシステムでは Cinderella で作成した図をシステム内に取り込むことが可能であるが、これまではその図を Javascript 側から操作することができなかった。よって、取り込んだ Cinderella の図がシステムから独立しており、Javascript や Algebrite で作成した問題に追従した図を作成できない状態であった。

この問題を解決するために、Cinderella を操作するためのプログラミング言語 CindyScript と Javascript の間の連携を図るために次の Cinderella 上の関数を準備した。

- `getjs(n,csvarname,jsvarname)` : Javascript 上の変数 `jsvarname` の値を Cinderella 上の変数 `csvarname` にコピーする。
- `setjs(n,csvarname,jsvarname)` : Cinderella 上の変数 `csvarname` の値を Javascript 上の変数 `jsvarname` にコピーする。
- `exejs(commandstring)` : 文字列変数 `commandstring` の値を Javascript の命令として実行する。
- `execs(commandstring)` : 文字列変数 `commandstring` の値を CindyScript の命令として実行する。
- `gettextarea(textareaid)` : 文字列変数 `textareaid` の名前をもったブラウザ上の `textarea` の中身を文字列として返す。
- `exectextarea(textareaid)` : 文字列変数 `textareaid` の名前をもったブラウザ上の `textarea` の中身を CindyScript の命令をして実行する。
- `console(n,string)` : n 番目のコンソール (ブラウザ上の `textarea`) に文字列変数 `string` の値を書き込む。
- `console(n,string)` : n 番目のコンソール (ブラウザ上の `textarea`) に変数 `point` の名前を持つ Cinderella の点の座標を書き込む。

また、同様に Javascript 上の関数として次のものを準備した。

- `getcs(n,jsvarname,csvarname)` : Cinderella 上の変数 `csvarname` の値を Javascript 上の変数 `jsvarname` にコピーする。
- `setcs(n,jsvarname,csvarname)` : Javascript 上の変数 `jsvarname` の値を Cinderella 上の変数 `csvarname` にコピーする。

- `execs(n,cscommandstring)` : 文字列変数 `cscommandstring` の中身を、 n 番目の Cinderella の図に対する CindyScript の命令として実行する。
- `gettextarea(textareaid)` : 文字列変数 `textareaid` の名前をもったブラウザ上の `textarea` の中身を文字列として返す。
- `exectextarea(textareaid)` : 文字列変数 `textareaid` の名前をもったブラウザ上の `textarea` の中身を Javascript の命令をして実行する。

上記の関数を用いたサンプルを示す。まず、Javascript から CindyScript へのデータの受け渡しが行われる例である。この例では次のように命令が実行される。

- (i) まず、図 3 の Javascript の命令が実行される。これは Javascript の変数 a, b, c に 2 以上 5 以下の乱数の整数を代入する命令である。
- (ii) 次に図 4 の CindyScript の命令 `getjs(1, 'a', 'a')` が実行される。これは上の (i) で Javascript の変数 a に代入された乱数の値を、CindyScript の変数 a に代入するものである。同様に、`getjs(1, 'b', 'b')`, `getjs(1, 'c', 'c')` は CindyScript の変数 b, c にも Javascript の変数 b, c の値 (これらには (i) で乱数が代入されている) を、CindyScript の b, c に代入する。
- (iii) 図 4 の CindyScript の命令 `plot(cos(a*x^2+b*x+c))` が実行される。これは Cinderella の図の中に $\cos(ax^2 + bx + c)$ のグラフをプロットするものである。

これらの命令が実行された結果を図 5 に示す。図 5 の関数 $f(x)$ はランダムに生成され、そのグラフがその下に表示されている。

次に逆に CindyScript から Javascript へのデータの受け渡しが行われる例を示す。この例は Cinderella から取り込んだ三角形の頂点を動かしたとき、その三角形の重心の軌跡を記録するものである。次のように動作する。

- (i) まず、図 6 の Cinderella の三角形の図の点 A, B, C を適当に動かす。
- (ii) そうすると、図 7 の Cinderella の命令が実行される (ここに書かれた命令は点 A, B, C が動いたときに自動的に実行されるようになっている)。この命令は 3 点 A, B, C の重心の座標を計算し、その座標が変化 (すなわち点 A, B, C が動いた) 時に、その座標を 2 番目のテキストボックス (図 7 の `con2` の名前のもの) に書き出す命令である。これにより、重心の座標の軌跡がテキストボックス内に書き込まれていく。

実行結果は図 8 である。このように、図 6 の点 A, B, C を動かすと三角形 ABC の重心の軌跡が図 8 の `con2` という名前のテキストボックスに書き込まれていく。図 8 の「output」をクリックすると、書き込まれた軌跡がファイルに出力されるようになっている。

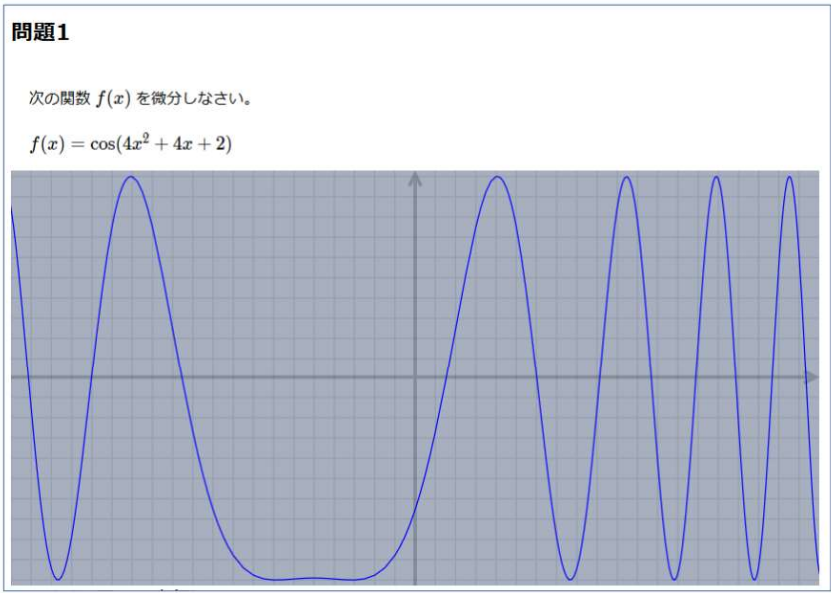


図 5 : 実行画面

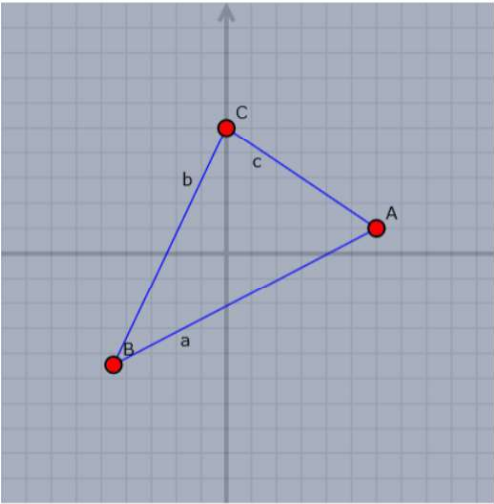


図 6 : Cinderella から取り込んだ三角形

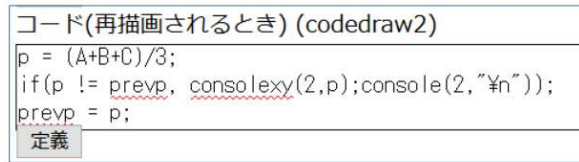


図 7 : CindyScript の命令

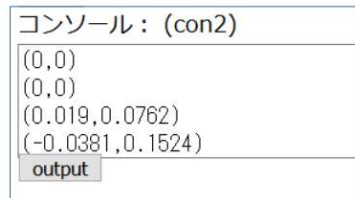


図 8 : 実行画面

4 おわりに

Javascript を活用した教材作成システムについて述べた。このシステムは ITC を活用した教材作成を容易にするためのものであり、教材作成者としては現場の教員を想定している。ブラウザ上で動作するため、パソコン・スマホ・タブレットで動作し、教材作成をブラウザ上で行うことが可能である。また、Javascript の命令を埋め込むことができるので、様々な処理が行え、数式処理機能を持っているため簡易的な E-Learning 教材を作成することもできる。

また、Cinderella で作成した図を取り込み、CindyScript と Javascript を用いてその図を操作したり、その図からデータを取得することが可能である。本稿では、その機能を活用した例を 2 つ挙げた。

今後は更に機能強化を行うとともに、教材の作成例を増やしていきたい。

参考文献

- [1] Moodle official homepage : URL <https://moodle.org/>
- [2] Moodle official homepage : STACK Plugin Description
URL https://moodle.org/plugins/qtype_stack
- [3] Algebrite official homepage : URL <http://algebrite.org/>
- [4] Quill official homepage : URL <https://quilljs.com/>

- [5] T. Kitamoto, M. Kanako, S. Takato : “E-learning system with Computer Algebra based on JavaScript programming language”, *Proc. of ATCM 2018*, pp. 123–233, Yogyakarta, 2018.