

# JavaScriptプログラミングによる インタラクティブなホームページの構築

## Construction of an interactive home page by programming on JavaScript

神保 雅人

**要 旨** JavaScriptはJava言語と異なり、直接HTML文書の中に書き込める。これを用いて、ホームページを閲覧しているユーザの操作で作動するプログラムを作成することが出来る。本稿では、JavaScriptプログラミングによる関数電卓の構築を通じて、インタラクティブなホームページの構築方法について論ずる。

### 1. はじめに

ホームページ作成方法に関する昨今の変化は実にめまぐるしいものである。文献 1) はHTMLに関して、サン・マイクロシステムズ社が監修して昨年出版されたものであるが、この時点では、ユーザの操作によって変化する、インタラクティブなホームページ作成にはCGI (Common Gateway Interface) が用いられていた。

CGIの利用には、CまたはUNIX上のスクリプト言語Perlによるプログラミングを必要とする。それに対して、機種に依存しないプログラミングを目指してサン・マイクロシステムズ社が開発した言語Java<sup>2)</sup>が、昨年より俄かに注目を浴びる様になった。

これはネットスケープ社が、動的なホームページ作りに利用しようという趣旨で、Javaアプレット(小規模なアプリケーションを意味する造語)にNetscape Navigatorを対応させたことによる。

ネットスケープ社は更に、それまで自社開発していたスクリプト言語の文法をJavaに揃え、JavaScriptを開発した。このJavaScriptにより、容易にインタラクティブなホームページ作成が行える様になった。本稿ではJavaScriptプログラミングによる関数電卓の構築と、そのホームページ上への実装について論ずる。

## 2. JavaScriptの特徴

### 2. 1. JavaとC

近年、開発用言語の標準がC及びC++となって以来、新たに設計された言語はすべて、これらを参考にしている。Javaの場合には、オブジェクト指向の言語という点で、文法はC++に近い。両者の特筆される相違として、Java言語で作成されたプログラムは機種に依存せずに動作可能である、ということが挙げられる。

これは、JavaコンパイラはJava仮想マシンを想定したJavaバイトコードを生成し、実行時にはJavaインタプリタにより、Javaバイトコードが解釈・実行される、という2段階のパスに分離されて実現されている。

C	Java
・宣言された配列の範囲を超えたメモリが参照可能	・コンパイルエラーまたはランタイムエラーとなる
・ポインタを用いて、任意のメモリ領域にアクセス可能	・ポインタ演算は存在しない
・確保した動的メモリは、解放する操作を行う必要がある	・メモリの解放は自動的に行われる

表1. CとJavaとの安全性に関する対比

また、Javaでは機能的には制限を受けても、実行時に有害なコードからシステムを守る、安全性に力点を置いて設計されている。この点に関してJavaとCとを対比すると、表1のようになる。この特長により、Javaがインターネット上で利用される様になったのである。

### 2. 2. JavaScriptとJava

Javaの場合には、通常言語と同様なプログラミングの他に、アプレットと呼ばれるプログラムを作成しておいて、それをHTMLの<APPLET>タグ<sup>3)</sup>により呼び出す、という利用の仕方が可能である。

それに対して、JavaScriptは、HTMLファイルに<SCRIPT>タグ<sup>3)</sup>により、ソースコードが埋め込まれて用いられる、スクリプト言語である。JavaScriptの文法は、上述した様に、Javaに似通った仕様となっている。<sup>4)</sup>

Java	JavaScript
・コンパイル済みのアプレットがクライアント側で実行される	・ソースコードがクライアント側で解釈・実行される
・オブジェクト指向言語	・オブジェクトベース言語（オブジェクトの継承なし）
・HTMLとは分離している	・HTMLによるGUIの挙動を制御可能

表2. JavaとJavaScriptとの特徴に関する対比

なお、JavaとJavaScriptとの主な相違点を列挙すると、表2の様になる。JavaScriptの動作環境には、大別してnavigatorオブジェクトとwindowオブジェクトとが用意されており、またその下層にframe, document, history等のオブジェクトが用意されている。<sup>4)</sup> JavaScriptによるコーディングはこれらを利用して行う。また、現時点に於けるWWWブラウザとしては、Netscape Navigatorの他に、Internet ExplorerがJavaScriptに対応している。

### 3. JavaScriptによる関数電卓の構築

#### 3. 1. 関数電卓の設計

JavaScriptによって関数電卓を構築するにあたり、仕様を以下の様に決定する。これは、MS-Windowsに付属する関数電卓ユーティリティよりも、市販のプログラム可能な関数電卓に近い操作性を目指すものである。

- 1) マウス操作で数式を入力し、それを計算式表示用ボックスに表示する。
- 2) 計算式が確定したら、等号(=)ボタンをマウスでクリックすることにより、計算結果表示用ボックスに数値で計算結果を表示する。
- 3) 直前の入力を取り消す為の「CE」ボタンを設ける。
- 4) 電卓のメモリ機能をシミュレートする為の非表示ボックスを設け、ストア、リコール、プラス、マイナス、クリアに相当するボタン(MS, MR, M+, M-, MC)を設置する。
- 5) 計算式表示用ボックス、計算結果表示用ボックス、及びメモリ機能用非表示ボックスの内容を同時に消去する為の「C」ボタンを設ける。

- 6) 定数として、円周率 ( $\pi$ ) 及び自然対数の底 (E) のボタンを設ける。
- 7) 16進数入力用のボタン「Hex」, 及び「a」～「f」のボタンを設ける。

実際に計算式表示用ボックスに表示される数式の記述方法は、JavaScriptの文法に従う。これはCの文法に近く、数値の表記法はCと同様である。例えば16進数は0x ffの様に、先頭に「0x」を付ける。また、8進数には「0」を付け、077の様に表す。従って、ここでは10進数と16進数、8進数との間は切り替える方式を採らず、混合演算が可能な仕様とする。

また、C同様に四則演算の他に、剰余を求める演算子「%」がJavaScriptでも利用できるので、「mod」ボタンのクリックにより、計算式表示用ボックスには「%」が入力される仕様とする。(通常の電卓のパーセント表示との混同を避ける意味で、「%」ボタンは設けない。)

なお、計算結果表示用ボックスの表示は17桁とする。これは、正の数の場合に小数点以下15桁の表示を可能とする為の仕様である。実際の計算はより多くの桁数を求め、計算結果表示用ボックスに値を返しているのので、このボックスをクリックした後にカーソルを右に移動すれば、残りの桁を参照することは可能である。

### 3. 2. 関数電卓のコーディング

ここでは先ず、JavaScriptのソースコードを埋め込んだHTMLのリストを与え、後に関数電卓の構築方法について論ずる。

#### リスト 1. Mathcalc.html

```
<HTML>
<HEAD>
<!--
      Copyright (c) Masato JIMBO 1996
      All Rights Reserved
-->

<TITLE>JavaScript Calculator by Masato Jimbo</TITLE>

      <SCRIPT LANGUAGE="JavaScript">
      <!--
          function clear_all(formula, answer, temp) {
```

## リスト1. (続き1)

```
        formula.value = "";  
        answer.value = "";  
        temp.value = "";  
    }  
  
    function insert_char(formula, character, temp) {  
        temp.value = formula.value;  
        formula.value += character;  
    }  
  
    function erase_char(formula, temp) {  
        formula.value = temp.value;  
    }  
  
    function eval_formula(formula, answer) {  
        with (Math) {  
            if (eval(formula.value) > 1.0e-15  
                || eval(formula.value) < -1.0e-15) {  
                answer.value = eval(formula.value);  
            }  
            else  
            {  
                answer.value = 0;  
            }  
        }  
    }  
  
    function mem_clear(dummy) {  
        dummy.value = "";  
    }  
  
    function mem_store(answer, dummy) {  
        dummy.value = answer.value;  
    }  
  
    function mem_recall(formula, dummy) {  
        if (eval(dummy.value) < 0) {  
            formula.value += "(" + dummy.value + " )";  
        }  
        else  
        {  
            formula.value += dummy.value;  
        }  
    }  
}
```

## リスト1. (続き2)

```
function mem_plus(answer, dummy) {
    if (dummy.value == "") {
        edv = 0;
    }
    else
    {
        edv = eval(dummy.value);
    }
    dummy.value = edv + eval(answer.value);
}

function mem_minus(answer, dummy) {
    if (dummy.value == "") {
        edv = 0;
    }
    else
    {
        edv = eval(dummy.value);
    }
    dummy.value = edv - eval(answer.value);
}
```

// --> JavaScript 対応のブラウザでご覧ください。

</SCRIPT>

</HEAD>

<BODY BGCOLOR="#7777AA">

<H2><FONT COLOR="#FFFF55">JavaScript による関数電卓</FONT> by Masato JIMBO

</H2><P>

<FORM>

<FONT COLOR="#00FFBB">計算式 : </FONT><INPUT TYPE="text" VALUE="" NAME="formula" SIZE=50 ><P>

<INPUT TYPE="button" VALUE=" ( " onClick="insert\_char(this.form.formula, '(' , this.form.temp)">

<INPUT TYPE="button" VALUE=" )" onClick="insert\_char(this.form.formula, ')', this.form.temp)">

<INPUT TYPE="button" VALUE=" m o d " onClick="insert\_char(this.form.formula, '%', this.form.temp)">

<INPUT TYPE="button" VALUE=" H e x " onClick="insert\_char(this.form.formula, '0x', this.form.temp)">

<INPUT TYPE="button" VALUE=" M C " onClick="mem\_clear(this.form.dummy)">

### リスト 1. (続き 3)

```
<INPUT TYPE="button" VALUE=" C " onClick="clear_all(this.form.formula,
this.form.answer,this.form.temp)">
<INPUT TYPE="button" VALUE=" C E " onClick="erace_char(this.form.formula,
this.form.temp)">
<INPUT TYPE="button" VALUE=" = " onClick="eval_formula(this.form.formula,
this.form.answer)">
<BR>
<INPUT TYPE="button" VALUE=" a s i n" onClick="insert_char(this.form.formula,
'asin(',this.form.temp)">
<INPUT TYPE="button" VALUE=" s i n" onClick="insert_char(this.form.formula,
'sin(',this.form.temp)">
<INPUT TYPE="button" VALUE=" e x p" onClick="insert_char(this.form.formula,
'exp(',this.form.temp)">
<INPUT TYPE="button" VALUE=" M R " onClick="mem_recall(this.form.formula,
this.form.dummy)">
<INPUT TYPE="button" VALUE=" a " onClick="insert_char(this.form.formula,'a',
this.form.temp)">
<INPUT TYPE="button" VALUE=" d " onClick="insert_char(this.form.formula,'d',
this.form.temp)">
<INPUT TYPE="button" VALUE=" 7 " onClick="insert_char(this.form.formula,'7',
this.form.temp)">
<INPUT TYPE="button" VALUE=" 8 " onClick="insert_char(this.form.formula,'8',
this.form.temp)">
<INPUT TYPE="button" VALUE=" 9 " onClick="insert_char(this.form.formula,'9',
this.form.temp)">
<INPUT TYPE="button" VALUE=" / " onClick="insert_char(this.form.formula,'/',
this.form.temp)">
<BR>
<INPUT TYPE="button" VALUE=" a c o s" onClick="insert_char(this.form.formula,
'acos(',this.form.temp)">
<INPUT TYPE="button" VALUE=" c o s" onClick="insert_char(this.form.formula,
'cos(',this.form.temp)">
<INPUT TYPE="button" VALUE=" p o w" onClick="insert_char(this.form.formula,
'pow(',this.form.temp)">
<INPUT TYPE="button" VALUE=" M S " onClick="mem_store(this.form.answer,
this.form.dummy)">
<INPUT TYPE="button" VALUE=" b " onClick="insert_char(this.form.formula,'b',
this.form.temp)">
<INPUT TYPE="button" VALUE=" e " onClick="insert_char(this.form.formula,'e',
this.form.temp)">
<INPUT TYPE="button" VALUE=" 4 " onClick="insert_char(this.form.formula,'4',
this.form.temp)">
<INPUT TYPE="button" VALUE=" 5 " onClick="insert_char(this.form.formula,'5',
this.form.temp)">
```

## リスト 1. (続き 4)

```
<INPUT TYPE="button" VALUE="6" onClick="insert_char(this.form.formula,'6',
this.form.temp)">
<INPUT TYPE="button" VALUE="*" onClick="insert_char(this.form.formula,'*',
this.form.temp)">
<BR>
<INPUT TYPE="button" VALUE="atan" onClick="insert_char(this.form.formula,
'atan(',this.form.temp)">
<INPUT TYPE="button" VALUE="tan" onClick="insert_char(this.form.formula,
'tan(',this.form.temp)">
<INPUT TYPE="button" VALUE="abs" onClick="insert_char(this.form.formula,
'abs(',this.form.temp)">
<INPUT TYPE="button" VALUE="M+" onClick="mem_plus(this.form.answer,
this.form.dummy)">
<INPUT TYPE="button" VALUE="c" onClick="insert_char(this.form.formula,'c',
this.form.temp)">
<INPUT TYPE="button" VALUE="f" onClick="insert_char(this.form.formula,'f',
this.form.temp)">
<INPUT TYPE="button" VALUE="1" onClick="insert_char(this.form.formula,'1',
this.form.temp)">
<INPUT TYPE="button" VALUE="2" onClick="insert_char(this.form.formula,'2',
this.form.temp)">
<INPUT TYPE="button" VALUE="3" onClick="insert_char(this.form.formula,'3',
this.form.temp)">
<INPUT TYPE="button" VALUE="-" onClick="insert_char(this.form.formula,'-',
this.form.temp)">
<BR>
<INPUT TYPE="button" VALUE="sqrt" onClick="insert_char(this.form.formula,
'sqrt(',this.form.temp)">
<INPUT TYPE="button" VALUE="log" onClick="insert_char(this.form.formula,
'log(',this.form.temp)">
<INPUT TYPE="button" VALUE="max" onClick="insert_char(this.form.formula,
'max(',this.form.temp)">
<INPUT TYPE="button" VALUE="M-" onClick="mem_minus(this.form.answer,
this.form.dummy)">
<INPUT TYPE="button" VALUE="π" onClick="insert_char(this.form.formula,'PI',
this.form.temp)">
<INPUT TYPE="button" VALUE="E" onClick="insert_char(this.form.formula,'E',
this.form.temp)">
<INPUT TYPE="button" VALUE="O" onClick="insert_char(this.form.formula,'0',
this.form.temp)">
<INPUT TYPE="button" VALUE="." onClick="insert_char(this.form.formula,',',
this.form.temp)">
<INPUT TYPE="button" VALUE="," onClick="insert_char(this.form.formula,',',
this.form.temp)">
```



## リスト 1. (続き 5)

```
<INPUT TYPE="button" VALUE="+" onCLICK="insert_char(this.form.formula,'+',
this.form.temp)">
<P>

<FONT COLOR="#FF7722">計算結果 : </FONT><INPUT TYPE="text" VALUE="" NAME="answer"
SIZE=17 ><P>

<INPUT TYPE="hidden" VALUE="" NAME="temp" SIZE=10 ><P>
<INPUT TYPE="hidden" VALUE="" NAME="dummy" SIZE=17><P>
</FORM>

</BODY>
</HTML>
```

この関数電卓は主に、`<SCRIPT>` ~ `</SCRIPT>` タグ間に記述された、イベントハンドラとしての関数の定義と、`<FORM>` ~ `</FORM>` タグ間に埋め込まれた「`button`」オブジェクト、及びそれに付随する「`onClick`」イベント（ボタンがクリックされた際に発生するイベント）とで構成されている。

また、計算式表示ボックス及び計算結果表示用ボックスは、共に「`text`」オブジェクトの埋め込みで与えられ、それぞれのボックスの名前は「`formula`」及び「`answer`」と名付けてある。なお、「`hidden`」オブジェクトの埋め込みで与えられ、「`temp`」及び「`dummy`」と名付けたボックスは共に非表示で、それぞれ「`CE`」ボタン及びメモリ機能を実現する為のものである。

なお、`<SCRIPT>` ~ `</SCRIPT>` タグ間の内容は、このタグに対応していないWWWブラウザの利用者の為に、このタグが無視された場合にHTMLのコメントの扱い（「`<!--`」 ~ 「`-->`」で挟む）となる様、設定している。その際、「`JavaScript`対応のブラウザをご覧ください。」の一文は、`JavaScript`の動作に支障を来さない様、`JavaScript`のコメント行として、`//`を付している。（これもC++と同様。）

個々のイベントハンドラ側で、呼び出し元のフォームの部品を参照するには、部品があるフォームオブジェクトへの参照を意味する「`this.form`」という引数を受け取る必要がある。ここでは、複数のテキストボックスを扱う為に、イベントハンドラの引数も複数個あるので、引数は「`this.form.formula`」の様になる。

定義されているイベント、イベントハンドラ及びボタンの対応は、表3の様になる。`JavaScript`では型は固定的なものではなく、数値と文字列を加えると全体が文字列となる。従って、計算式の評価には、文字列として記述されている式を評価する関数「`eval`」を用いる必要がある。また、これはメモリ用ボックスへの計算結果の加算・減算

の際にも同様なことが言える。

イベント	イベントハンドラ	イベントを発生させるボタン
計算式表示ボックスへの文字列挿入及び退避用ボックス (temp) への文字列の退避	<code>insert_char</code>	( ) a s i n a c o s a t a n s q r t m o d s i n c o s t a n l o g H e x e x p p o w a b s m a x a ~ f π E 0 ~ 9 . , / * - +
計算式の評価及び計算結果表示用ボックスへの数値の表示	<code>eval_formula</code>	=
計算式表示ボックスへの直前の入力の取り消し	<code>erase_char</code>	CE
計算式表示ボックス, 計算結果表示用ボックス及び退避用ボックスの内容の消去	<code>clear_all</code>	C
メモリ用ボックス (dummy) への計算結果の格納	<code>mem_store</code>	MS
メモリ用ボックスから計算式表示用ボックスへの値の呼び出し	<code>mem_recall</code>	MR
メモリ用ボックスへの計算結果の加算	<code>mem_plus</code>	M+
メモリ用ボックスへの計算結果の減算	<code>mem_minus</code>	M-
メモリ用ボックス内の数値の消去	<code>mem_clear</code>	MC

表3. 関数電卓中のイベント, イベントハンドラ及びボタンの対応

なお, 「`eval_formula`」の中に, 「`with (Math)`」とあるが, これはJavaScriptで算術計算用にMathオブジェクトが用意されており, 定数はプロパティ, 算術関数はメソッドとして定義されていることに由来する。

これらは通常, 「`Math.E`」, 「`Math.sin(n)`」の様に参照される。この様にオブジェクト名を冠するのは煩雑であると共に, 計算式の表示の際にユーザには理解しがたいものとなる。そこで, 関数定義の最初に「`with (Math)`」を付けることにより, 「`eval_formula`」を呼び出す際には, 「`Math`」が陽に付されていなくても, これが付加されて解釈される。

また、イベントハンドラ中で「`formula.value`」等とあるのは、テキストボックスにはフィールドの内容を参照する為のプロパティとして`value`が定義されているので、これを利用して値の代入を行っている。

その他、幾つかのイベントハンドラ中に、精度の調整やエラー処理、及びメモリ用ボックスからの負の数値の呼び出しに対応する目的で「`if`」文を入れてある。この記述の仕方はCと同様である。

HTMLの`JavaScript`関係以外の部分は、骨格と題名を除けば、背景色と文字の大きさ及び色の指定のみである。

### 3. 3. 関数電卓の実装

上述の関数電卓のコーディングに際しては、以下の開発環境を利用した。

- 1) ハードウェア： 富士通製ノートパソコン FMV-BIBLO (5133 NA2/W) (CPU: Pentium 133MHz; RAM: 16+32MB)
- 2) OS: Microsoft Windows 95 Operating System
- 3) エディタ： 秀丸エディタ (シェアウェア)
- 4) ブラウザ： Netscape Navigator Version 2.02 I [ja]

デバッグはブラウザで出来栄を確認しながらエディタでソースを修正する、という方法で行った。ここでブラウザとして、`Netscape Navigator`を選んだ理由は、元々`JavaScript`がネットスケープ社によって開発されたものであることによる。

マイクロソフト社の`Internet Explorer`を用いれば、ソース表示の際に「メモ帳」を開くことにより、直ちにソースの修正が可能であるが、`JavaScript`に関しては、見栄えが`Netscape Navigator`のものとは異なる。

実際にパーソナルコンピュータ上でブラウザを用いて、この関数電卓を表示した例を図1及び図2に与える。前者は`Netscape Navigator`、後者は`Internet Explorer`による表示である。



図 1. Netscape Navigator 2.0.2.1 [JA] による表示

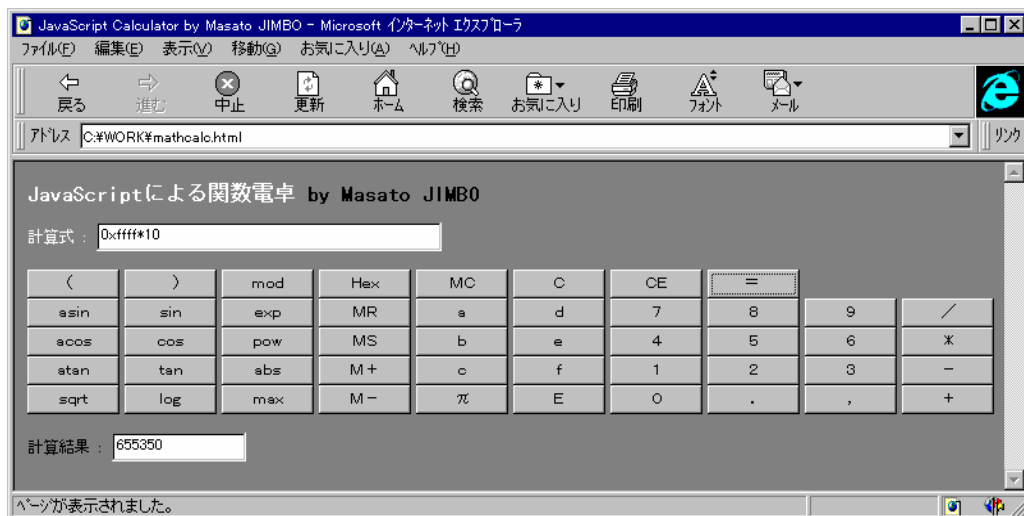


図 2. Internet Explorer 3.0 日本語版による表示

この様に、パーソナルコンピュータ上では動作確認済みである。本学では、筆者がインストールして起動している `httpd` (セキュリティの為、名称は秘匿事項) が、ワークステーション上に存在するので、このソースファイル「`Mathcalc.html`」を `ftp` で転送し、筆者が作成済みのホームページ (セキュリティの為、格納されているディレクトリ名は秘匿事項) の何処かに次の一行を加えれば、実装は完了となる。

```
<A HREF="/Mathcalc.html">J a v a S c r i p tによる関数電卓</A>
```

## 4. おわりに

本学では平成7年9月よりワークステーションを導入し、TRAIN経由で専用線によるインターネット利用が可能となった。それに伴い、本年度は筆者のゼミナールでは、1年生向けには `LaTeX` ソースの作成と、その電子メールによる交換・加工を、また2年生向けにはホームページの作成を主題にしている。

本稿の `JavaScript` によるインタラクティブなホームページは、その先に位置するものである。学生が `JavaScript` を理解するには、最低限 `C` の学習が必要であると考えられるが、幸いにも本学の情報系の学生は2年次に `C` を学習している。

この点に関して、本学では来年度よりプログラミングの授業が半期ずつ後ろにずれるので、従来は1年次に `COBOL` を学習していた科目の内容を `C` に変更することが、今後のネットワークとマルチメディア中心の情報化社会に対応させるには、望ましい姿である。

なお、本稿で扱った `JavaScript` は `HTML` のソースに直接埋め込む為に、プログラミングは容易であるが、必ずソースコードを公開することになる。本稿の関数電卓のソースコードに関しても例外ではなく、WWW上でアクセスしたユーザが自由に読み取ることが出来てしまう。従って、この関数電卓のホームページ上への実装の完了による公開は、本稿の本学紀要への掲載が受理されて引用が可能となる以前には行えない。

## 参考文献

- 1) Morris, M.E.S., *HTML for Fun and Profit*, Mountain View: Sun Soft Press (A Prentice Hall Title), 1995.
- 2) Flanagan, D., *Java in a Nutshell – A Desktop Quick Reference for Java Programmers*, O'Reilly & Associates, 1996 (邦訳：日本サン・マイクロシステムズ株式会社 監訳，永松健司；オライリー・ジャパン，1996) .
- 3) 株式会社アネク，HTMLタグ辞典，翔泳社，1996 (初版第1刷) .
- 4) 田中ひろゆき，はじめてのJava & JavaScriptプログラミング，ソフトバンク，1996 (初版) .