

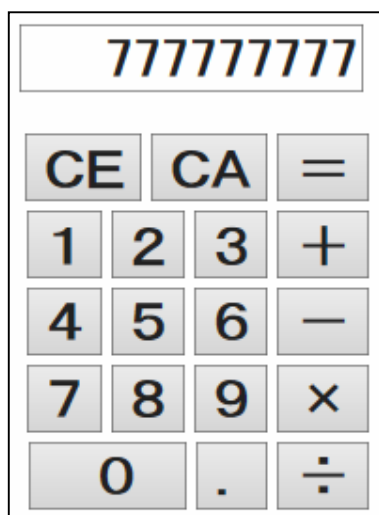
ななちゃんのIT教室

JavaScriptで電卓プログラムを作ろうの巻

by nara.yasuhiro@gmail.com

電卓アプリの課題でつまずいていた
JavaScript 入門者ななちゃんが立ち直るまでのお話

第 0.6 版 2017 年 5 月 7 日



もくじ

- 第1回 画面をデザインしよう
- 第2回 JavaScript 部分を書く
- 第3回 結果を入力欄に戻す
- 第4回 「=」キーを設ける
- 第5回 input2 欄 と 演算子表示欄 を 削る
- 第6回 「×」と「÷」を追加
- 第7回 テンキーボタン入力に
- 第8回 「クリア」キーを設ける、文字フォントを指定する
- 第9回 table を用いて整列する、小数点キーを設ける
- 第10回 連続計算対応に
- 発展課題
 - 二進数電卓(整数版)
 - 二進数電卓(小数対応版)
 - N進数電卓(小数対応版)

第1回 画面をデザインしよう

なな: JavaScript の電卓プログラムを作る課題ができなくて、自信をなくしてしまったの。

先生: 電卓プログラムって、足す/引く/掛ける/割る だけなので、簡単と思ってしまうけど、いろいろ細かい工夫が入っていて、実は結構複雑な動きをするのよ。表示欄が入力中の数の表示と、計算結果の表示を兼ねていたりして。

なな: どこからはじめたらいいのやら……

先生: まず、プログラムファイルの、枠組みを用意するの。どんなプログラムでも共通の部分なので、「template.html」のような名前のファイルで用意しておいて、コピーして使うと良いのよ。1文字間違えただけでも、ちゃんと動作しなくなることが多いから。



フリー素材
http://freeillustration.net



フリー素材
http://freeillustration.net

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <script>
    </script>
  </body>
</html>
```

そして、今回は、HTML の部分を使って、画面を作りましょう。まずは、「電卓」ということは忘れて、足し算、引き算をすることだけを考えましょう。

ここには
全角の
空白が
いくつか
入って
いる

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="input2">input2<p>
    <input type="text" id="operator" size="1" > )<input type="text" id="input1">input1<br>
    -----<br>
    <input type="text" id="output">output<p>
    <input type="button" value="+" onclick=tasu()>
    <input type="button" value="-" onclick=hiku()>
    <script>
    </script>
  </body>
</html>
```

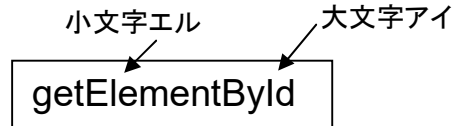
こんな画面表示になります。まだ JavaScript 部分が作ってないので、「+」や「-」のボタンをクリックしても、何も起こらないけど。

第2回 JavaScript 部分を書く

なな: 今度は、JavaScript の部分を書くのね。

先生: そうね。「document.getElementById('inputのid)」は、HTML の <input> 枠と、JavaScript の橋渡しをするものなの。これでもらったデータを、「input1p」などの変数に記憶しておいて、「... = input1p.value」のようにすると、input枠内に書き込んだデータを JavaScript で使ったり、「input1p.value = ...」のようにすると、input枠内に JavaScript からデータを書き込んだりできます。

なな: 「input」と言っても、入力も、出力もできるのね。



先生: 「電卓らしさ」こだわらなければ、これで完成よ。電卓のむずかしさは、電卓のサイズを小さくするために、数字表示欄をひとつにまとめて、いろいろな表示を切り替えるところにあるのよ。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="input2">input2<p>
    <input type="text" id="operator" size=1> )<input type="text" id="input1">input1<br>
    -----<br>
    <input type="text" id="output">output<p>
    <input type="button" value="+" onclick=tasu()>
    <input type="button" value="-" onclick=hiku()>
    <script>
    var input1p = document.getElementById('input1');
    var operatorp = document.getElementById('operator');
    var input2p = document.getElementById('input2');
    var outputp = document.getElementById('output');

    function tasu() {
      operatorp.value = "+";
      outputp.value = Number(input2p.value) + Number(input1p.value);
    }
    function hiku() {
      operatorp.value = "-";
      outputp.value = Number(input2p.value) - Number(input1p.value);
    }
    </script>
  </body>
</html>

```



このように、「1」、「2」のデータを入力して「+」ボタンをクリックすると、

output 欄に、足し算結果の「3」が表示される。「+」マークも左に表示される。

第3回 結果を入力欄に戻す

先生: まず、「電卓化」の第一歩は、計算結果の「output」欄のデータを、input1 欄にコピーすることなの。これで、1回の計算の結果を、次の計算の入力として使えるようになるし、output 欄を削るための「布石」になるの。



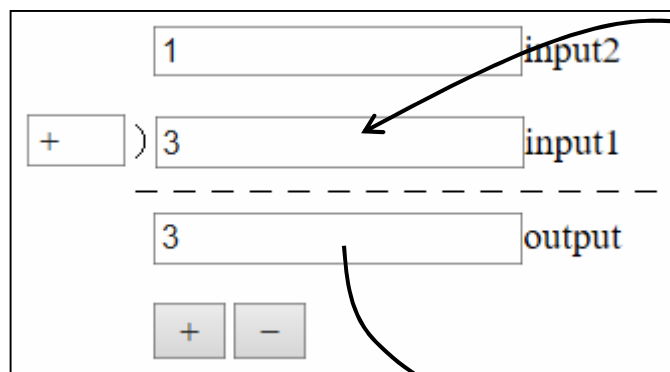
なな: 最終的に、数字欄をひとつにまとめることの第一歩として、output 欄を削除する準備をするのね。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="input2">input2<p>
    <input type="text" id="operator" size=1> )<input type="text" id="input1">input1<br>
    -----<br>
    <input type="text" id="output">output<p>
    <input type="button" value="+" onclick=tasu()>
    <input type="button" value="-" onclick=hiku()>
    <script>
var input1p  = document.getElementById('input1');
var operatorp = document.getElementById('operator');
var input2p  = document.getElementById('input2');
var outputp  = document.getElementById('output');

function tasu() {
  operatorp.value = "+";
  outputp.value = Number(input2p.value) + Number(input1p.value);
  input1p.value = Number(input2p.value) + Number(input1p.value);
}
function hiku() {
  operatorp.value = "-";
  outputp.value = Number(input2p.value) - Number(input1p.value);
  input1p.value = Number(input2p.value) - Number(input1p.value);
}
    </script>
  </body>
</html>

```



データを
コピー
する


先生： 次に、計算結果は input1 欄に表示されるので、省略可能な output 欄を削除しましょう。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="input2">input2<p>
    <input type="text" id="operator" size="1" ><input type="text" id="input1">input1<br>
    -----<br>
    <input type="button" value="+" onclick=tasu(>
    <input type="button" value="-" onclick=hiku(>
    <script>
var input1p  = document.getElementById("input1");
var operatorp = document.getElementById("operator");
var input2p  = document.getElementById("input2");

function tasu() {
  operatorp.value = "+";
  input1p.value = Number(input2p.value) + Number(input1p.value);
}
function hiku() {
  operatorp.value = "-";
  input1p.value = Number(input2p.value) - Number(input1p.value);
}
    </script>
  </body>
</html>

```



<input type="text" id="output">
output<p> を削除した

input2

)

input1

「1」と「2」を入力して、「+」ボタンをクリックする

input2

)

input1

足した結果の「3」が input1 欄に表示される。
この結果を使って、次の計算を行える。



第4回 「=」キーを設ける

先生: 「1 + 2 =」という操作にしましょう。 + / - の入力時に input1→input2 にデータを移動します。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="input2">input2<p>
    <input type="text" id="operator" size=1> ) <input type="text" id="input1">input1<br>
    -----<br>
    <input type="button" value="+" onclick=tasu(>
    <input type="button" value="-" onclick=hiku(>
    <input type="button" value="=" onclick=equal(>
    <script>
    var input1p  = document.getElementById('input1');
    var operatorp = document.getElementById('operator');
    var input2p  = document.getElementById('input2');

    function tasu() {
      operatorp.value = "+";
      input2p.value = input1p.value;
      input1p.value = "";
    }
    function hiku() {
      operatorp.value = "-";
      input2p.value = input1p.value;
      input1p.value = "";
    }
    function equal() {
      if (operatorp.value == "+") {
        input1p.value = Number(input2p.value) + Number(input1p.value);
      }
      else if (operatorp.value == "-") {
        input1p.value = Number(input2p.value) - Number(input1p.value);
      }
      operatorp.value = "=";
    }
    </script>
  </body>
</html>

```



input2

) 1 input1

+ - =

①「1」を入力してから「+」をクリック ↑

1 input2

+) input1

+ - =

②「1」が input2 欄に移動した ↑

1 input2

+) 2 input1

+ - =

③「2」を入力してから「=」をクリック ↑

1 input2

=) 3 input1

+ - =

④計算結果の「3」が input1 欄に表示された

第5回 input2 欄 と 演算子表示欄 を 削る


先生: input2 欄は、ユーザが直接入力する必要がなくなったので、省略可能です。省略可能といっても、表示が不要になるということで、機能は残す必要があるの。演算子表示欄も、表示しなくても良いわね。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="operator" size="1" > <input type="text" id="input1">input1<br>
    -----<br>
    <input type="button" value="+" onclick=tasu(>
    <input type="button" value="-" onclick=hiku(>
    <input type="button" value="=" onclick=equal(>
    <script>
var input1p = document.getElementById('input1');
var operatorp = document.getElementById('operator');
var input2;

function tasu() {
  operatorp.value = "+";
  input2 = input1p.value;
  input1p.value = "";
}
function hiku() {
  operatorp.value = "-";
  input2 = input1p.value;
  input1p.value = "";
}
function equal() {
  if (operatorp.value == "+") {
    input1p.value = Number(input2) + Number(input1p.value);
  }
  else if (operatorp.value == "-") {
    input1p.value = Number(input2) - Number(input1p.value);
  }
  operatorp.value = "=";
}
    </script>
  </body>
</html>
    
```

<input type="text" id="input2">
input2<p> を削除した



)

input1

+

-

=



演算子表示欄を削除して、<input type=text> は、input1 ひとつだけに

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type=text id=input1><p>
    <input type=button value="+" onclick=tasu()>
    <input type=button value="-" onclick=hiku()>
    <input type=button value="=" onclick=equal()>
    <script>
var input1p = document.getElementById("input1");
var operator;
var input;

function tasu() {
  operator = "+";
  input2 = input1p.value;
  input1p.value = "";
}
function hiku() {
  operator = "-";
  input2 = input1p.value;
  input1p.value = "";
}
function equal() {
  if (operator == "+") {
    input1p.value = Number(input2) + Number(input1p.value);
  }
  else if (operator == "-") {
    input1p.value = Number(input2) - Number(input1p.value);
  }
  operator = "=";
}
    </script>
  </body>
</html>

```

3		
+	-	=



第6回 「×」と「÷」を追加

先生：「×」と「÷」の機能を追加しましょう。演算子登録部分は、似たようなものが4つ並ぶことになるので、まとめてしまいましょう。

なな：これで、キーボード入力タイプの電卓は完成ね！



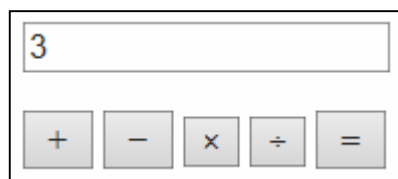
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="input1"><p>
    <input type="button" value="+" onclick=kigou("+")>
    <input type="button" value="-" onclick=kigou("-")>
    <input type="button" value="*" onclick=kigou("*")>
    <input type="button" value="/" onclick=kigou("/")>
    <input type="button" value="=" onclick=equal()>
    <script>
var input1p = document.getElementById("input1");
input1p.focus();
var operator;
var input;

function kigou(k) {
  operator = k;
  input2 = input1p.value;
  input1p.value = "";
  input1p.focus();
}
function equal() {
  if (operator == "+") { input1p.value = Number(input2) + Number(input1p.value); }
  else if (operator == "-") { input1p.value = Number(input2) - Number(input1p.value); }
  else if (operator == "*") { input1p.value = Number(input2) * Number(input1p.value); }
  else if (operator == "/") { input1p.value = Number(input2) / Number(input1p.value); }
  operator = "=";
  input1p.focus( );
}
    </script>
  </body>
</html>

```

focus() は、枠を選択状態にして、すぐに入力できるようにするもの



第7回 テンキーボタン入力に

先生： 今度は、電卓らしく、数字のボタンをクリックするようにしましょう。数字の表示も、電卓のように右寄せにしましょう。ボタンに対応する 1 桁の数字を、現在表示されている内容の右端にくっつけるの。くっつけるのに、「+」を使います。「+」は、数字だと足し算、文字だと接続をするの。

なな： わあ！ ずいぶん電卓らしくなったわね！



```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <input type="text" id="input1" style="text-align:right"><p>
    <input type="button" value="=" onclick=equal()><p>
    <input type="button" value="1" onclick=kazu("1")>
    <input type="button" value="2" onclick=kazu("2")>
    <input type="button" value="3" onclick=kazu("3")>
    <input type="button" value="+" onclick=kigou("+")><br>
    <input type="button" value="4" onclick=kazu("4")>
    <input type="button" value="5" onclick=kazu("5")>
    <input type="button" value="6" onclick=kazu("6")>
    <input type="button" value="-" onclick=kigou("-")><br>
    <input type="button" value="7" onclick=kazu("7")>
    <input type="button" value="8" onclick=kazu("8")>
    <input type="button" value="9" onclick=kazu("9")>
    <input type="button" value="x" onclick=kigou("x")><br>
    <input type="button" value=" 0 " onclick=kazu("0")>
    <input type="button" value="÷" onclick=kigou("÷")><p>
    <script>
var input1p = document.getElementById("input1");
var operator;
var input;

function kazu(n) {
  if (input1p.value == "0") input1p.value = "";
  input1p.value = input1p.value + n;
}
function kigou(k) {
  operator = k;
  input2 = input1p.value;
  input1p.value = "";
}
function equal() {
  if (operator == "+") { input1p.value = Number(input2) + Number(input1p.value); }
  else if (operator == "-") { input1p.value = Number(input2) - Number(input1p.value); }
  else if (operator == "x") { input1p.value = Number(input2) * Number(input1p.value); }
  else if (operator == "÷") { input1p.value = Number(input2) / Number(input1p.value); }
  operator = "=";
}
    </script>
  </body>
</html>

```

右寄せ

+ / -
より
x / ÷
が少し
小さい

Number(...) は、文字列を
数字に変換します

Number がないと、「+」が
結合になってしまいます。
(1 + 2 → 12)

x 等では
Number
は省略可

第8回 「クリア」キーを設ける、文字フォントを指定する

先生： 今度は、「CE」(クリアテントリ)キーと、「CA」(クリアオール)キーを追加するわね。

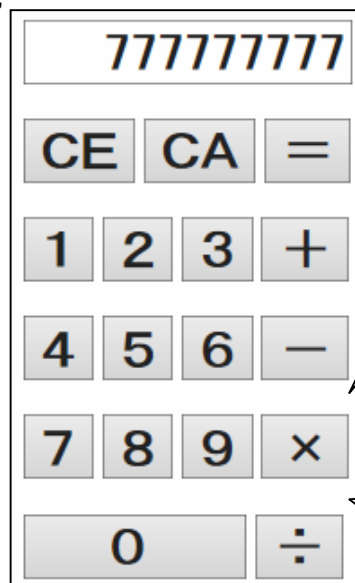
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      input { font-family: "MS Pゴシック", sans-serif; font-weight:600; font-size:x-large; }
    </style>
  </head>
  <body>
    <input type=text id=input1 size=13 style="text-align:right"><p>
    <input type=button value="CE" onclick=clr()>
    <input type=button value="CA" onclick=clrAll()>
    <input type=button value="=" onclick=equal()><p>
    <input type=button value="1" onclick=kazu("1")>
    <input type=button value="2" onclick=kazu("2")>
    <input type=button value="3" onclick=kazu("3")>
    <input type=button value="+" onclick=kigou("+")><p>
    <input type=button value="4" onclick=kazu("4")>
    <input type=button value="5" onclick=kazu("5")>
    <input type=button value="6" onclick=kazu("6")>
    <input type=button value="-" onclick=kigou("-")><p>
    <input type=button value="7" onclick=kazu("7")>
    <input type=button value="8" onclick=kazu("8")>
    <input type=button value="9" onclick=kazu("9")>
    <input type=button value="x" onclick=kigou("x")><p>
    <tr><td><input type=button value=" 0 " onclick=kazu("0")>
      <input type=button value="÷" onclick=kigou("÷")><p>    <script>
var input1p  = document.getElementById('input1');
var operator, input;

function clr() { input1p.value = ""; }

function clrAll() {
  input1p.value = "";
  input2 = "";
  operator = "";
}
function kazu(n) {
  if (input1p.value == "0") input1p.value = "";
  input1p.value = input1p.value + n;
}
function kigou(k) {
  operator = k;
  input2 = input1p.value;
  input1p.value = "";
}
function equal() {
  if (operator == "+") { input1p.value = Number(input2) + Number(input1p.value); }
  else if (operator == "-") { input1p.value = Number(input2) - Number(input1p.value); }
  else if (operator == "x") { input1p.value = Number(input2) * Number(input1p.value); }
  else if (operator == "÷") { input1p.value = Number(input2) / Number(input1p.value); }
  operator = "=";
}
    </script>
  </body>
</html>

```



+ / -
と
× / ÷
が同じ
大きさ

少し
横位置
が
ずれて
いる

第9回 table を用いて整列する、小数点キーを設ける

先生：HTMLの「表」の機能を使って、ボタン配列を整えましょう。

なな：「小数点」キーも追加するのね。



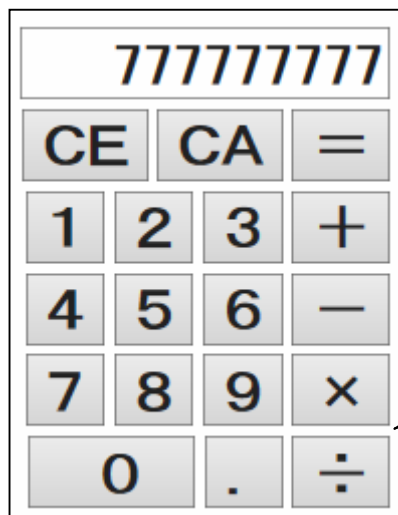
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      input { font-family: "メイリオ", sans-serif; font-weight:600; font-size:x-large; }
      table { text-align: right }
    </style>
  </head>
  <body>
    <table>
      <tr><td><input type=text id=input1 size=13 readonly
        style="text-align:right"></td></tr>
      <tr><td><input type=button value="CE" onclick=clr()>
        <input type=button value="CA" onclick=clrAll()>
        <input type=button value="" onclick=equal()></td></tr>
      <tr><td><input type=button value="1" onclick=kazu("1")>
        <input type=button value="2" onclick=kazu("2")>
        <input type=button value="3" onclick=kazu("3")>
        <input type=button value="+" onclick=kigou("+")></td></tr>
      <tr><td><input type=button value="4" onclick=kazu("4")>
        <input type=button value="5" onclick=kazu("5")>
        <input type=button value="6" onclick=kazu("6")>
        <input type=button value="-" onclick=kigou("-")></td></tr>
      <tr><td><input type=button value="7" onclick=kazu("7")>
        <input type=button value="8" onclick=kazu("8")>
        <input type=button value="9" onclick=kazu("9")>
        <input type=button value="x" onclick=kigou("x")></td></tr>
      <tr><td><input type=button value="0" onclick=kazu("0")>
        <input type=button value="." onclick=kazu(".")>
        <input type=button value="÷" onclick=kigou("÷")></td></tr>
    </table>
  </body>
</html>

```


右寄せ

キーボード入力不可に



右寄せによって、
右端が揃っている

先生：「1.2.3」みたいに、小数点が 2 つ以上入力されてしまわないように対策をしましょう。



```

<script>
var input1p  = document.getElementById('input1');
var operator;
var input;
var shousuu = 0;

function clr() {
  input1p.value = "";
  shousuu = 0;
}
function clrAll() {
  input1p.value = "";
  input2 = "";
  operator = "";
  shousuu = 0;
}
function kazu(n) {
  if (operator == "=") {
    operator = "";
    input1p.value = "";
  }
  if (n == '.') {
    if (shousuu == 1) return;
    if (input1p.value == "") input1p.value = "0.";
    else
      input1p.value = input1p.value + ".";
    shousuu = 1;
    return;
  }
  if (input1p.value == "0") input1p.value = "";
  input1p.value = input1p.value + n;
}
function kigou(k) {
  operator = k;
  input2 = input1p.value;
  input1p.value = "";
}
function equal() {
  if (operator == "+") { input1p.value = Number(input2) + Number(input1p.value); }
  else if (operator == "-") { input1p.value = Number(input2) - Number(input1p.value); }
  else if (operator == "x") { input1p.value = Number(input2) * Number(input1p.value); }
  else if (operator == "÷") { input1p.value = Number(input2) / Number(input1p.value); }
  operator = "=";
}
}
</script>
</body>
</html>

```

小数点が 2 つ
入力されるのを
防止

「.01」を「0.01」に変える

第10回 連続計算対応に

先生： 連続計算対応に発展させましょう。

入力:	1 + 2 + 3 =
出力:	11 2 3 3 6



なな: 「=」キーを押すのを省略して、+ / - / × / ÷ キーを押して、計算を続ける機能ね。計算の途中結果を消さない工夫が必要なのね。「1+2+」のところで、表示が消えずに、途中結果の「3」を表示しておいて、次の数字入力があった時に、はじめて、表示をクリアするのね！

```

<script>
var input1p = document.getElementById('input1');
var operator;
var input;
var shousuu = 0;
var toBeErased = 0;

function clr() {
  input1p.value = "";
  shousuu = 0;
}
function clrAll() {
  input1p.value = "";
  input2 = "";
  operator = "";
  shousuu = 0;
}
function kazu(n) {
  if (toBeErased == 1) {
    toBeErased = 0;
    input1p.value = "";
  }
  if (n == '.') {
    if (shousuu == 1) return;
    if (input1p.value == "") input1p.value = "0.";
    else
      input1p.value = input1p.value + ".";
    shousuu = 1;
    return;
  }
  if (input1p.value == "0") input1p.value = "";
  input1p.value = input1p.value + n;
}
function kigou(k) {
  if (operator != "") { equal(); }
  operator = k;
  input2 = input1p.value;
  toBeErased = 1;
}
function equal() {
  if (operator == "+") { input1p.value = Number(input2) + Number(input1p.value); }
  else if (operator == "-") { input1p.value = Number(input2) - Number(input1p.value); }
  else if (operator == "x") { input1p.value = Number(input2) * Number(input1p.value); }
  else if (operator == "÷") { input1p.value = Number(input2) / Number(input1p.value); }
  operator = "";
  toBeErased = 1;
}
</script>
</body>
</html>

```



演算子が連続したら
「=」処理を呼び出して
から、演算子登録

発展課題

自分だけの、

「POW」キー（「2」、「POW」、「3」、「=」の順に入力すると 2^3 の計算結果、「2」、「POW」、「0.5」、「=」の順に入力すると $\sqrt{2}$ の計算結果を表示）(Math.pow(x,y) を利用できます)、

「BMI」キー（体重、BMIキー、身長、= キー の順に入力すると肥満度を表示）、

などを追加してみましょう。

小数点以下 何桁まで表示するかを変更できるようにしてみましょう。

ヒント：100 倍してから整数化し、100 で割ると、小数点以下 2 桁になります。

二進数電卓(整数版)

二進数の加減乗除計算ができる電卓です。「10 + 10 + 10 = 110」のような連続計算ができます。割り算で小数点以下の桁が出る場合がありますが、連続計算では小数点以下が切りすてられます。

```

<!DOCTYPE html>
<html>
  <head><meta charset="utf-8">
    <style type="text/css"> input { font-size:150%; }</style>
  </head><body>
    <input type="text" id="disp" value="0" readonly style="text-align: right;"><p>
    <input type="button" onclick=num(1) value="1">
    <input type="button" onclick=num(0) value="0"><p>
    <input type="button" onclick=opr("+") value="+">
    <input type="button" onclick=opr("-") value="-">
    <input type="button" onclick=opr("*") value="*">
    <input type="button" onclick=opr("/") value="/"><p>
    <input type="button" onclick=cal() value="=">
    <input type="button" onclick=clearE() value="CE">
    <input type="button" onclick=clearA() value="C"><p>
    <script>
var mem = 0, op = "", clr = 0;
var dispp = document.getElementById("disp");

function num(n) {
  if (clr == 1) { clr = 0; dispp.value = 0; }
  if (dispp.value == 0) { dispp.value = ""; }
  dispp.value = ""+ dispp.value + n;
}

function opr(o) {
  if ((op != "") && (op != "=")) cal();
  mem = dispp.value; clr = 1; op = o;
}

function cal() {
  if (op == "=") { return; }
  dispp.value = (eval("parseInt(mem,2)"
    +op+"parseInt(dispp.value,2)")).toString(2);
  mem = 0; op = "="; clr = 1;
}

function clearE() { dispp.value = 0; }

function clearA() { mem = 0; op = ""; clearE(); }
</script>
</body>
</html>

```

新入力→表示クリア

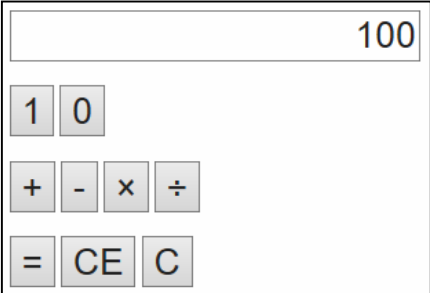
01 や 00 にならないように

連続計算

二進数として評価

計算結果を二進数に

二進数として評価



二進数電卓(小数対応版)

二進数の加減乗除の計算ができる電卓です。小数点以下のある数の入力や計算ができます。連続計算もできます。(二進数にも小数点数がある！)

```

<!DOCTYPE html>
<html><head>
  <meta charset="utf-8">
  <style> input { font-size:150%; }</style>
</head><body>
  <input type=text id=disp value=0 readonly style="text-align: right;"><p>
  <input type=button onclick=num(1) value="1">
  <input type=button onclick=num(0) value="0">
  <input type=button onclick=pnt() value=".">
  <input type=button onclick=cal() value="="><p>
  <input type=button onclick=opr("+") value="+">
  <input type=button onclick=opr("-") value="-">
  <input type=button onclick=opr("*") value="x">
  <input type=button onclick=opr("/") value="÷"><p>
  <input type=button onclick=clearE() value="CE">
  <input type=button onclick=clearA() value="C"><p>
  <script>
var mem = 0, op = "", clr = 0, dpt = 0;
var dispp = document.getElementById("disp");

function pnt() {
  if (dpt == 1) return;
  dpt = 1;
  if (clr == 1) { clr = 0; dispp.value = 0; }
  dispp.value = "" + dispp.value + ".";
}
function num(n) {
  if (clr == 1) { clr = 0; dispp.value = 0; }
  if ((dpt == 0) && (dispp.value == 0)) { dispp.value = ""; }
  dispp.value = "" + dispp.value + n;
}
function opr(o) {
  if ((op != "") && (op != "=")) cal();
  mem = dispp.value; clr = 1; op = o; dpt = 0;
}
function cal() {
  if (op == "=") { return; }
  dispp.value = (eval("parseBinary(mem,2)"
    +op+"parseBinary(dispp.value,2)")).toString(2);
  mem = 0; op = "="; clr = 1; dpt = 0;
}
function clearE() { dispp.value = 0; dpt = 0; }
function clearA() { mem = 0; op = ""; clearE(); }

function parseBinary(s) {
  var v = 0, p = 0, w = 0.5;
  for (var i in s) {
    if (s[i] == ".") { p = 1; }
    else if (p == 0) { v = v*2 + Number(s[i]); }
    else { v = v + Number(s[i])*w; w /= 2; }
  }
  return v;
}
</script>
</body>
</html>

```

0.01

1	0	.	=
+	-	x	÷
CE	C		

小数点二重入力防止

二進数文字列解釈

小数点

整数部

N進数電卓(小数対応版)

基数を任意に変更できる、加減乗除の計算ができる電卓です。小数点以下のある数の入力や計算もできます。

```

<!DOCTYPE html>
<html><head>
  <meta charset="utf-8">
  <style> input { font-size:150%; }</style>
</head><body>
  基数: <input type="text" size=3 value=2 onchange="radix=this.value;init()"><br>
  <input type="text" id="disp" value=0 readonly style="text-align: right;"><p>
  <input type="button" onclick=pnt() value=".">
  <input type="button" onclick=cal() value="=">
  <input type="button" onclick=clearE() value="CE">
  <input type="button" onclick=clearA() value="C"><p>
  <input type="button" onclick=opr("+") value="+">
  <input type="button" onclick=opr("-") value="-">
  <input type="button" onclick=opr("*") value="x">
  <input type="button" onclick=opr("/") value="÷"><p>
  <div id="area"><input type="button" value="0" onclick=num(this.value)></div>
</script>
var radix = 2, mem, op, clr, dpt;
var dispp = document.getElementById("disp");
var areap = document.getElementById("area");
var nd = areap.children[0];
function init() {
  areap.innerHTML = "";
  for (var i=0; i<radix; i++) {
    var cln = nd.cloneNode(true); cln.value = i.toString(radix);
    areap.appendChild(cln);
    if (i % 4 == 3) areap.appendChild(document.createElement("br")); }
  clearA();
} init();
function pnt() {
  if (dpt == 1) return;
  dpt = 1;
  if (clr == 1) { clr = 0; dispp.value = 0; }
  dispp.value = "" + dispp.value + "."; }
function num(n) {
  if (clr == 1) { clr = 0; dispp.value = 0; }
  if ((dpt == 0) && (dispp.value == 0)) { dispp.value = ""; }
  dispp.value = "" + dispp.value + n; }
function opr(o) {
  if ((op != "") && (op != "=")) cal();
  mem = dispp.value; clr = 1; op = o; dpt = 0; }
function cal() {
  if (op == "=") { return; }
  dispp.value = (eval("parseNumber(mem,radix)"
    +op+"parseNumber(dispp.value,radix)")).toString(radix);
  mem = 0; op = "="; clr = 1; dpt = 0; }
function clearE() { dispp.value = 0; dpt = 0; }
function clearA() { mem = 0; op = ""; clr = 0; clearE(); }
function parseNumber(s) {
  var v = 0, p = 0, w = 1/radix;
  for (var i in s)
    if (s[i] == ".") { p = 1; }
    else if (p == 0) { v = v*radix + sToN(s[i]); }
    else { v = v + sToN(s[i])*w; w /= radix; }
  return v; }
function sToN(s) {
  if (s.charCodeAt(0) >= "a".charCodeAt(0)) return s.charCodeAt(0) - "a".charCodeAt(0) + 10;
  else return s.charCodeAt(0) - "0".charCodeAt(0); }
</script>
</body>
</html>

```

基数が変わったらリセット

「0」ボタンを記憶

0~基数-1 ボタンとしてコピー

4 つごとに改行



N 進数表現文字列を内部表現数に

「a」~は 10~と解釈