

ななちゃんのIT教室

メニューを極めよう の巻

by nara.yasuhiro@gmail.com

ななちゃんが、メニューの
使い方をマスターするという お話

第 0.1 版 2017 年 6 月 23 日



フリー素材
<http://freeillustration.net>



いらすとやフリー素材
<http://www.irasutoya.com/>

もくじ

- 第1回 秘密道具:マイ・コンソール2
- 第2回 チェックボックスとラジオボタン
- 第3回 セレクトボックスの基本的な使い方
- 第4回 セレクトボックスの情報参照
- 第5回 セレクトボックスの生成
- 第6回 オプショングループ
- 第7回 複数選択型
- 第8回 応用例

第1回 秘密道具:マイ・コンソール2

なな: クリじい、「メニュー」の勉強をするんだけど、便利な秘密道具はない?

クリ: あるぞ、あるぞ。定番秘密道具の「マイ・コンソール2」。マイ・コンソールを発展させたんじゃ。



マイコンソール Ver. 2

```
<input type=text id=in1 value="abc">
```

HTML定義

①ここに HTML の命令を書きこむ。複数行でも良い

```
abc
```

②「HTML」定義ボタンをクリック

③HTML 部品が現れる

```
var in1 = document.getElementById("in1");
in1.value;
```

JavaScript の命令「log()」で、出力することもできる

イベントハンドラから JavaScript の命令「write()」で、出力することもできる

実行

④ここに JavaScript の命令を書きこむ。複数行でも良い

⑤実行ボタンをクリック

```
<= var in1 = document.getElementById("in1");
in1.value;
=> String string "abc"
```

システムからのメッセージ

⑥命令が転記 (<=) され、エラーメッセージ、結果の「値」(=>)、log()、write() の結果が表示される

出力例

上記④で書き込んだ内容がここに転記される (<= が印)

```
<= var in1 = document.getElementById("in1");
in1.value;
=> String string "abc"
```

結果の「型」と「値」(=> が印)

```
var in1 = document.getElementById("in1");
in1.value; // String string "abc"
```

本教材ではこのように圧縮表示しています

JavaScript 命令/式

実行結果の「型」と「値」

「○; ○」のように、複数の JavaScript 命令がある場合、一番右の命令の型、値だけ表示される

```

<!DOCTYPE html>
<html> <head>
  <meta charset="utf-8">
  <title>コンソール</title>
</head> <body>
  <h3>マイコンソール Ver. 2</h3>
  <textarea rows="10" cols="80" id=htmlIn><input type=text id=in1 value="abc">
</textarea>
  <br><input type=button onClick=htmlDef() value="HTML 定義"><br>
  <div id=htmlOut></div><br>
  <textarea rows="19" cols="80" id=pg autofocus>
var in1 = document.getElementById("in1");
in1.value;</textarea>
  <br><input type=button onClick=go() value="実行">
  <br>システムからのメッセージ
  <br><textarea rows="20" cols="80" id=log></textarea>
  <script>
var geval = eval, logd;
var logp = document.getElementById("log");
var pgp  = document.getElementById("pg");
var htmlIn = document.getElementById("htmlIn");
var htmlOut = document.getElementById("htmlOut");

function htmlDef() {
  htmlOut.innerHTML = htmlIn.value; }
function clog(s) { logp.value += s; }
function log(s) { logd += s; }
function write(s) { logp.value += s; }
function typels(obj) {
  return(Object.prototype.toString.call(obj).slice(8, -1)); }
function isPrimitive(x) {
  return (typeof x)!="object"; }
function toLiteral(x) {
  if (typels(x)=="Number" && isNaN(x)) return "NaN";
  if (x === Infinity) return "Infinity";
  if ((typels(x)!="Symbol")&&(-x === Infinity)) return "-Infinity";
  if (typels(x)=="Set") return "Set("+JSON.stringify([...x])+")";
  if (typels(x)=="Map") return "Map("+JSON.stringify([...x])+")";
  return JSON.stringify(x); }
function type(x) { return "" + (typeof x); }
function isInteger(n) { return n%1 === 0; }
function keys(obj) { return Object.keys(obj); }
function go() {
  logd = "";
  try {
    var v = geval(pgp.value);
    clog("<= " + pgp.value + "\n=> "
      + typels(v) + " " + type(v) + " " + toLiteral(v) + "\n");
    pgp.value = "";
    logp.scrollTop = logp.scrollHeight;
    pgp.focus();
  } catch(e) { clog("<= " + pgp.value + "\n=>! " + e + "\n");
    pgp.value = "";
    logp.scrollTop = logp.scrollHeight;
    pgp.focus();
  }
  if (logd != "") clog(logd + "\n");
}
</script>
</body>
</html>

```

第2回 チェックボックスとラジオボタン

なな: まず、チェックボックスの使い方を教えてください。

先生: HTML と、そのチェック状態を調べる JavaScript は、下記のようになります。各項目は独立で複数選択可。

```
<input type="checkbox" id="cb1"> チェックボックス 1<br>
<input type="checkbox" id="cb2"> チェックボックス 2<br>
<input type="checkbox" id="cb3"> チェックボックス 3<br>
```

```
var cb = [0,0,0], cbs = [0,0,0];
for (var i in cb) {
  cb[i] = document.getElementById('cb'+(Number(i)+1));
  cbs[i] = cb[i].checked;
}
cbs; // Array object [false,true,false]
```

チェックボックス1
 チェックボックス2
 チェックボックス3



[false,true,false] というのは、チェックボックス2 が選択されているという意味です。

なな: 部分をクリックする必要があるけど、文字の部分でも良いようにできる? それから、はじめにチェックボックス2 を選択状態にしておくとか。

先生: HTML 部分を下記のように変えればOKです。

```
<label><input type="checkbox" id="cb1"> チェックボックス 1</label><br>
<label><input type="checkbox" id="cb2" checked=""> チェックボックス 2</label><br>
<label><input type="checkbox" id="cb3"> チェックボックス 3</label><br>
```

ラジオの
戦局
イメージ

なな: ラジオボタンは?

先生: 下記のようなプログラムになります。name でグループ化すると、グループ内の一つしか選択できません。

```
<input type="radio" id="rb1" value="rb_1" name=g1>ラジオボタン1<br>
<input type="radio" id="rb2" value="rb_2" name=g1>ラジオボタン2<br>
<input type="radio" id="rb3" value="rb_3" name=g2>ラジオボタン3<br>
<input type="radio" id="rb4" value="rb_4" name=g2>ラジオボタン4<br>
```

```
var rb = [0,0,0,0], rbs = [0,0,0,0];
for (var i in rb) {
  rb[i] = document.getElementById('rb'+(Number(i)+1));
  rbs[i] = rb[i].checked;
}
rbs; // Array object [false,true,true,false]
```

ラジオボタン1
 ラジオボタン2
 ラジオボタン3
 ラジオボタン4



なな: value= の値は、JavaScript から読めるの? 初期選択や、文字列部分のクリックを可能にするのは?

先生: 下記のようにします。value 値を表示し、選択 (*) と非選択 (0) のマーク付けをしました。

```
<label><input type="radio" id="rb1" value="rb_1" name=g1>ラジオボタン1</label><br>
<label><input type="radio" id="rb2" value="rb_2" name=g1 checked="">ラジオボタン2</label><br>
<label><input type="radio" id="rb3" value="rb_3" name=g2 checked="">ラジオボタン3</label><br>
<label><input type="radio" id="rb4" value="rb_4" name=g2>ラジオボタン4</label><br>
```

```
var rb = [0,0,0,0], rbs = [0,0,0,0];
for (var i in rb) {
  rb[i] = document.getElementById('rb'+(Number(i)+1));
  rbs[i] = (rb[i].checked?"(*)":"(0)") + rb[i].value;
}
rbs; // Array object ["(0)rb_1","(*)rb_2","(*)rb_3","(0)rb_4"]
```



第3回 セレクトボックスの基本的な使い方

なな: 次は、セレクトボックスね。基本的な使い方からお願いします。

<pre><select id=sb> <option value=op1>セレクト1</option> <option value=op2>セレクト2</option> <option value=op3>セレクト3</option> </select></pre>	
<pre>var sb = document.getElementById('sb'); var sbs = sb.value; sbs; // String string "op3"</pre>	
<pre>var sb = document.getElementById('sb'); var sbs = sb.selectedIndex + ":" + sb.options[sb.selectedIndex].value; sbs; // String string "2:op3"</pre>	

先生: HTML と JavaScript です。セレクト3 を選択後、JavaScript で状態を読み込みました。

なな: 選択した瞬間に JavaScript を呼び出すことはできるかしら？

先生: できます。3 つのプログラム例です。



<pre><select onchange=f(this.value)> <option value=op1>セレクト1</option> <option value=op2>セレクト2</option> <option value=op3>セレクト3</option> </select></pre>
<pre>function f(v) { write(v + "\n"); } // op3</pre>

<pre><select id=sb> <option value=op1>セレクト1</option> <option value=op2>セレクト2</option> <option value=op3>セレクト3</option> </select></pre>
<pre>var sb = document.getElementById('sb'); sb.addEventListener('change',function(e) { write(e.target.value + "\n"); }); // op3</pre>
<pre>var sb = document.getElementById('sb'); sb.addEventListener('change',function() { write(this.value + "\n"); }); // op3</pre>

なな: はじめに、デフォルト値として、セレクト2 を選択状態にできるかしら？

先生: 下記のように、selected をつけます。

<pre><select id=sb> <option value=op1>セレクト1</option> <option value=op2 selected>セレクト2</option> <option value=op3>セレクト3</option> </select></pre>



第4回 セレクトボックスの情報参照

なな: セレクトボックスのどんな情報が JavaScript で参照できるの？

先生: <select>タグや、<option>タグの内容を参照することができます。



```
<select id=sb>
  <option value=op1>セレクト1</option>
  <option value=op2>セレクト2</option>
  <option value=op3 id=item3>セレクト3</option>
</select>
```

```
var sb = document.getElementById('sb');
sb; // HTMLSelectElement object {"0":{},"1":{},"item3":{}}
sb.type; // String string "select-one"
sb.length; // Number number 3
sb.options[0].text; // String string "セレクト1"
sb[0].text; // String string "セレクト1"
sb.item(0).text; // String string "セレクト1"
sb.namedItem("item3").value; // String string "セレクト3"
sb["item3"].text; // String string "セレクト3"
sb[2].text; // String string "セレクト3"
sb.outerHTML; // String string "<select id='sb'>\n
  <option value='op1'>セレクト1</option>\n
  <option value='op2'>セレクト2</option>\n
  <option id='item3' value='op3'>セレクト3</option>\n</select>"
sb.innerHTML; // String string "\n
  <option value='op1'>セレクト1</option>\n
  <option value='op2'>セレクト2</option>\n
  <option id='item3' value='op3'>セレクト3</option>\n"
sb[0].innerHTML; // String string "セレクト1"
sb[0].outerHTML; // String string "<option value='op1'>セレクト1</option>"
```



第5回 セレクトボックスの生成

なな: HTML の <option> タグって、記述量が多いから、選択肢が多い時にプログラムが長くなるわね。例えば、日本の 47 都道府県 のメニューを作るのも大変そう。

先生: HTML では、空のタグを用意し、JavaScript で <option> を挿入することが可能です。都道府県だったら、名前を配列に入れて置いて、for ループでタグを作ることができます。まずは、基本から、

```
<select id=sb></select>

var sb = document.getElementById('sb');
sb.add(new Option('セレクト1', 'op1'));
sb.add(new Option('セレクト2', 'op2'));
sb.add(new Option('セレクト3', 'op3'));
```



これで、3 項目の option ができます。↑

```
var sb = document.createElement('select');
var option = document.createElement("option");
option.text = "東京";
option.value = 1;
sb.add(option);
sb.length++;
sb.options[sb.length - 1].value = 2;
sb.options[sb.length - 1].text = "横浜";
document.body.appendChild(sb);
```

これは、<select> タグまで JavaScript で生成し、<body> の末尾に書き込む例です。↑

```
var data = [['セレクト1', 'op1'], ['セレクト2', 'op2'], ['セレクト3', 'op3']];
var sb = document.getElementById('sb');
for (var i in data) sb.add(new Option(data[i][0], data[i][1]));
```

```
var data = [['セレクト1', 'op1'], ['セレクト2', 'op2'], ['セレクト3', 'op3']];
var sb = document.getElementById('sb');
for (var i in data) {
    var option = document.createElement("option");
    option.text = data[i][0];
    option.value = data[i][1];
    sb.add(option);
}
```



上記の 2 つは、配列データに基づいて <option> タグを生成する例です。都道府県の場合は、

```
var data = ["北海道", "青森県", "岩手県", "宮城県", "秋田県", "山形県", "福島県",
"茨城県", "栃木県", "群馬県", "埼玉県", "千葉県", "東京都", "神奈川県",
"新潟県", "富山県", "石川県", "福井県", "山梨県", "長野県", "岐阜県", "静岡県",
"愛知県", "三重県", "滋賀県", "京都府", "大阪府", "兵庫県", "奈良県", "和歌山県",
"鳥取県", "島根県", "岡山県", "広島県", "山口県",
"徳島県", "香川県", "愛媛県", "高知県",
"福岡県", "佐賀県", "長崎県", "熊本県", "大分県", "宮崎県", "鹿児島県", "沖縄県"];
var sb = document.getElementById('sb');
for (var i in data) sb.add(new Option(data[i], i));
```

こんな感じになります。

なな: <option> が 47 行ならばのより、はるかにスマートね！

なな: <option> を削除する方法はあるの？

先生: あります。下記の例を見てください。null を書き込む方法と、remove() を使う方法があります。何番目の <option> を削除するかを指定します。番号は 0 からはじまります。

```
<select id=sb>
  <option value=op1>セレクト1</option>
  <option value=op2>セレクト2</option>
  <option value=op3 id=item3>セレクト3</option>
</select>
```



```
var sb = document.getElementById('sb');
sb; // HTMLSelectElement object {"0":{},"1":{},"item3":{}}
sb.length; // Number number 3
sb.options[0] = null; // Null object null
sb.length; // Number number 2
sb; // HTMLSelectElement object {"0":{},"item3":{}}
sb.remove(1); // Undefined undefined undefined
sb.length; // Number number 1
sb; // HTMLSelectElement object {"0":{}}
```

削除

削除

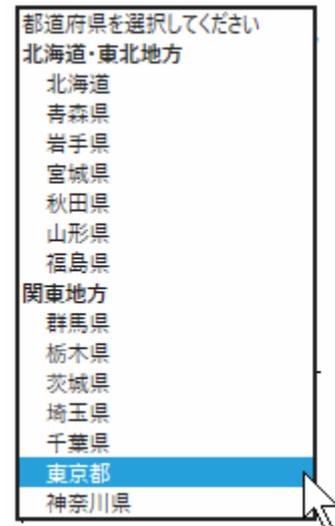


第6回 オプショングループ

なな: オプショングループって何。

先生: 例で説明するのが早そうね。下記の例では、「北海道・東北地方」がオプショングループ欄で、メニューを見やすくするけど、実際には選択できない項目になります。

```
<select name="pref">
  <option value="">都道府県を選択してください</option>
  <optgroup label="北海道・東北地方">
    <option value="北海道">北海道</option>
    <option value="青森県">青森県</option>
    <option value="岩手県">岩手県</option>
    <option value="宮城県">宮城県</option>
    <option value="秋田県">秋田県</option>
    <option value="山形県">山形県</option>
    <option value="福島県">福島県</option>
  </optgroup>
  <optgroup label="関東地方">
    <option value="群馬県">群馬県</option>
    <option value="栃木県">栃木県</option>
    <option value="茨城県">茨城県</option>
    <option value="埼玉県">埼玉県</option>
    <option value="千葉県">千葉県</option>
    <option value="東京都" selected="selected">東京都</option>
    <option value="神奈川県">神奈川県</option>
  </optgroup>
</select>
```

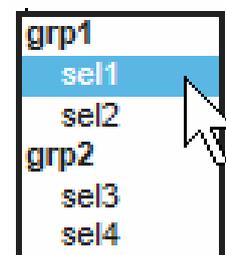


なな: これも、JavaScript で作れるの？

先生: はい。こんなプログラムになります。

```
<select id=sb></select>
```

```
var data = [['g','grp1', 'gp1'],['i','sel1', 'op1'],['i','sel2', 'op2'],
            ['g','grp2', 'gp2'],['i','sel3', 'op3'],['i','sel4', 'op4']];
var sb = document.getElementById('sb');
var gp = sb, option;
for (var i in data) {
  if (data[i][0] == 'g') {
    gp = document.createElement("optgroup");
    gp.label = data[i][1];
    sb.appendChild(gp);
  }
  else {
    option = document.createElement("option");
    option.text = data[i][1];
    option.value = data[i][2];
    gp.appendChild(option);
  }
}
```



第7回 複数選択型

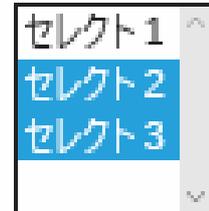
なな: 複数選択ができるメニューもあるの?

先生: はい。こんな感じになります。



```
<select id=sb multiple>
  <option value=op1>セレクト1</option>
  <option value=op2>セレクト2</option>
  <option value=op3>セレクト3</option>
</select>
```

```
var sb = document.getElementById('sb');
sb.type; // String string "select-multiple"
sb.selectedOptions.length; // Number number 2
sb.selectedOptions[0].value; // String string "op2"
sb.selectedOptions[0].text; // String string "セレクト2"
```



なな: 使ってみました。はじめから開いたメニューの形になるので、ctrl キーを押しながらクリックしてひとつずつ追加してゆく方法と、開始項目クリック、shift キーを押しながら終了項目クリックで連続領域を選択する方法があるのね。

先生: そうね。selectedOptions.length で、選択した項目の数が分かり、selectedOptions[n] で、選択した項目のひとつひとつにアクセスできます。



第8回 応用例

先生: 応用例です。項目を選択すると、対応するウェブページにジャンプします。

```
<form name="test">
<select name="linkselect" onchange="check()">
<option value="">選択して下さい</option>
<option value="http://www.yahoo.co.jp/">Yahoo!Japan</option>
<option value="http://www.google.co.jp/">Google</option>
</select>
</form>
```

```
function check(){
  var obj = document.test.linkselect;
  var index = obj.selectedIndex;
  if (index != 0) {
    href = obj.options[index].value;
    location.href = href;
  }
}
```

