テキストボックスの操作

ここから、フォームの情報を JavaScript で取得したり、 操作したりする方法について解説します。まずはテキストボックスの操作を考えます。

テキストボックスを操作するコード

まずはテキストボックスに入力された文字を取得したり、書き換えたりします。 HTML の body 内に以下のように記入してみて下さい。

```
<form name="js">
お名前：<input type="text" name="txtb" value=""><br>
<input type="button" value="歓迎１" onclick="tbox1()"><br>
<input type="button" value="消去" onclick="clr()"><br>
</form>

<script>

//テキストボックスの文字を取得する
function tbox1(){

  var str1=document.js.txtb.value;
  alert("ようこそ"+str1+"さん！");

}


//テキストボックスの文字を操作する
function clr(){

  document.js.txtb.value="";

}

</script>
```

既に何回か出てきましたが、 JavaScript でフォームを扱う時は form タグ， input タグに name 属性を付けます。 ここでは「js」と「txtb」という名称を付けました。

次に JavaScript 部分を見ていきます。 関数 tbox1()で、 変数 str1 を宣言し、 テキストボックスの文字列を取得して代入しています。 テキストボックスの文字列は以下の方法で取得できます。

document.form 名.input 名.value
テキストボックスに表示されている文字列を取得します
テキストエリア<textarea></textarea>の文字列も取得できます
最後に alert で取得した文字を加工し、出力するようにしています。
２つめの関数 clr()は、テキストボックスの文字を書き換えする関数です。
document.form.input.value を左辺に持ってくれば、 テキストボックスの value 値に右辺の文字列を代入することができます。 空の文字列を代入することで、テキストボックス内の文字列をクリアできます。

elements[]を利用したフォーム操作
JavaScript でフォームを操作する時には、 input タグの name 属性値を使う以外に elements[]を使う方法もあります。 elements[]を使うと、 フォームに設置された各ボックスやボタンを上から順番に取得することができます。 上記のフォームでは、1 番目がテキストボックス、2,3 番目がボタンとなります。
上記コードの続きに、以下のように書き足して試してみて下さい。

```
<script>

//elements[]を使って取得する
function tbox2(){

  var str2=document.js.elements[0].value;
  alert(str2+"さん、いらっしゃい！");

}

</script>

<form>
<input type="button" value="歓迎２" onclick="tbox2()"><br>
</form>
```

elements[]は配列になっているので、 最初のテキストボックスを取得する場合は elements[0]となります。配列なので、for 文を使って繰り返し操作する時などに便利です。

次のページでは、チェックボックスについて解説します。

チェックボックスを扱う

このページでは、チェックボックスを JavaScript で扱う方法について解説します。 どのチェックボックスが選択されているか確認したり、 どれもチェックされていないと警告を出したりしてみます。

チェックボックスを扱うスクリプト
HTML のボディ内に、以下のように記述してみてください。

```
<form name="chbox">
<p>あなたの好きな動物は？（複数可）</p>
<input type="checkbox" value="イヌ">イヌ<br>
<input type="checkbox" value="ネコ">ネコ<br>
<input type="checkbox" value="ウサギ">ウサギ<br>
<input type="checkbox" value="ハムスター">ハムスター<br>
<input type="checkbox" value="熱帯魚">熱帯魚<br>
<input type="checkbox" value="他">この中には無い<br>
<input type="button" value="確認" onclick="boxCheck()">
</form>

<script>

function boxCheck(){

 //チェックされた項目を記録する変数
 var str="";

 //for 文でチェックボックスを 1 つずつ確認
 for( i=0; i<6; i++ )
 {
  //チェックされているか確認する
  if( document.chbox.elements[i].checked )
  {
   //変数 str が空でない時、区切りのコンマを入れる
   if( str != "" ) str=str+",";

   //チェックボックスの value 値を変数 str に入れる
   str=str+document.chbox.elements[i].value;
  }
```

```
 }

 //str が空の時、警告を出す
 if( str=="" ){
    alert( "どれか選択してください。" );
 }else{
    alert( str + "が選択されました。" );
 }
}

</script>
```

チェックされているか確認する

では上記のスクリプトを見ていきましょう。 まずはフォーム部分ですが、JavaScript で扱うときは name 属性を指定するのが基本です。 ここでは「chbox」という名前を付けました。

次に関数 boxCheck()を作成します。 最初に、どの項目がチェックされているかを格納する変数 str を宣言します。

それから、チェックボックスを順番に確認していきます。 elements[]を使うとチェックボックスを配列として扱えるので、for 文と共に連続して確認できます。チェックボックスは６つあるので、配列の番号は 0～5 となります。 for 文の範囲は「i<6」にしましたが、勿論「i<=5」でも OK です。

各チェックボックスがチェックされているかどうかを確かめるには、checked を使います。以下の形式でチェック状態を取得します。

document.form 名.elements[].checked

チェックされていれば true、されていなければ false を取得します

if 文と組み合わせることによって、チェックされているか確認できます。

if (document.form の name 名.elements[].checked){ 処理 }

チェックされていれば、以降の処理が実行されます

チェックされている場合の処理について見ていきましょう。 チェックボックスは複数選択が可能です。 したがって変数 str に格納される値は、１つでは無い可能性もあります。 それで、複数指定された時はコンマで区切ってみました。 ただし最初だけはコンマを入れると変なので、変数 str の値が空の場合‘以外’にコンマを入れるようにしています。

変数 str に、チェックされているチェックボックスの value 値を格納していきます。 value 値の取得は、前項のテキストボックスと同じです。 変数 str に値を追加するので、str=str+…の形を取ります。

最後に alert を使って、選択されている項目を表示しています。 もしどれも選択されていけなければ変数 str は空ですから、警告を出します。
さて、上記のフォームでは「犬」と「この中には無い」を選ぶことが可能です。 これでは少し具合が悪いので、 「この中には無い」を選んだら他のチェックを外すようにしてみてはどうでしょうか？ 次のページでは、チェックマークを付けたり消したりする方法について解説します。
チェックの ON/OFF
前回はチェックボックスがチェックされているかどうかを確かめる方法について見ました。このページでは JavaScript でチェックを付けたり消したりする方法について考えます。
JavaScript でチェックを ON・OFF する
ここでは、前のページで見た HTML のフォームを利用していきます。 では JavaScript の部分に、以下の関数を書き足してみてください。

```
function noch(){
 //「この中には無い」がチェックされているか確認
 if( document.chbox.elements[5].checked )
 {
  //チェックされていたら、他の項目のチェックを外す
  for( i=0 ; i<5 ; i++)
  {
   document.chbox.elements[i].checked=false;
  }
 }
}
```

そして、関数 boxCheck()の先頭で、今書いた関数 noch()を呼び出すようにします。

```
function boxCheck(){

 noch();

 var str="";

 for(i=0;i<6;i++)
 {
  if(document.chbox.elements[i].checked)
  {
   if(str != "") str=str+",";
   str=str+document.chbox.elements[i].value;
```

```
  }
 }

 if(str==""){
   alert("どれか選択してください。");
 }else{
   alert(str+"が選択されました。");
 }
}
```

これで「この中には無い」がチェックされていたら、 他の項目のチェックを外すようにすることができます。 その後で各項目がチェックされているか確認することになります。

JavaScript コードの説明

では上記コードについて見ていきましょう。関数 noch()を宣言し、最初に if 文を使って「この中には無い」 がチェックされているかどうか確認しています。 「この中に無い」チェックボックスはフォームの中で 6 番目の要素です。 従って elements[]の大括弧内は 5（配列は 0 から数える）となります。

もしチェックされていたら、他の項目のチェックを消していきます。 for 文を使って、1 から 5 番目までの項目のチェックを連続して消します。 i の範囲は 0～4 となります。

次にこのページの要点にあたる、チェックマークの操作です。 チェックボックスのチェックを付けたり消したりするには、以下のように記述します。

document.form 名.elements[].checked = true/false

true ならチェックを付け、false なら消します。

最後に、関数 boxCheck()の冒頭で関数 noch()を呼び出します。 関数の中で他の関数を呼び出すのは初出かも知れませんが、このようなことも可能です。

ちなみに、チェック項目「この中には無い」を選択した瞬間に他の項目を消したい場合は、onchange イベントで関数 noch()を呼び出すとよいでしょう。

```
<input type="checkbox" value="他" onchange="noch()">この中には無い<br>
```

次のページでは、ラジオボタンの操作について考えます。 基本的にチェックボックスと操作は同じです。 でもそれではつまらないと思うので、ラジオボタンを使った 3 択クイズを作ってみたいと思います。

ラジオボタンを扱う

このページではフォームのラジオボタンを JavaScript で扱う方法について解説します。基本的にチェックボックスと扱い方は同じです。

ラジオボタンを使ったサンプル

ではラジオボタンを使って三択クイズを作ってみたいと思います。 HTML のボディ内に、以下のコードを記入してみてください。

```
<form name="radioB">
カナダの首都は？<br>
<input type="radio" name="Q1">オタワ<br>
<input type="radio" name="Q1">トロント<br>
<input type="radio" name="Q1">モントリオール<br>
<br>
スイスの首都は？<br>
<input type="radio" name="Q2">ジュネーブ<br>
<input type="radio" name="Q2">チューリッヒ<br>
<input type="radio" name="Q2">ベルン<br>
<br>
ドイツの首都は？<br>
<input type="radio" name="Q3">ハンブルク<br>
<input type="radio" name="Q3">ブレーメン<br>
<input type="radio" name="Q3">ベルリン<br>
<br>
スペインの首都は？<br>
<input type="radio" name="Q4">バルセロナ<br>
<input type="radio" name="Q4">マドリード<br>
<input type="radio" name="Q4">リスボン<br>
<br>
オーストラリアの首都は？<br>
<input type="radio" name="Q5">シドニー<br>
<input type="radio" name="Q5">メルボルン<br>
<input type="radio" name="Q5">キャンベラ<br>
<br>
<input type="button" value="採点" onclick="saiten()" />
</form>



<script>
function saiten(){
```

```
 var seikai=0; //正解数を入れる変数

 //答えの番号を配列に入れる
 var trueAns = new Array(0,5,8,10,14);

 //正解のラジオボタンがチェックされているか確認
 for (i=0 ; i<5 ; i++)
 {
   if( document.radioB.elements[trueAns[i]].checked )
     seikai++;
 }

 alert("あなたは"+seikai*20+"点でした！");
}
</script>
```

三択クイズの正解を見極める

では、上記コードについて調べていきましょう。 フォーム名は「radioB」にしました。 また各ラジオボタンの name 属性値は３つずつ同じにして、３つのうちからどれか１つ選ぶようにします。

続いて JavaScript の部分をみていきましょう。 まず関数 saiten()を作成します。 そして最初に正解数を入れる変数 seikai を宣言し、初期値 0 を代入しておきます。

次に正しい答えの番号を、配列 trueAns に入れていきます。 elements[]を使うので、1 番目のラジオボタンであれば 0，6 番目であれば 5 になります。

それから for 文を使って、 正しい答えのラジオボタンのみ調べていきます。 配列 trueAns の要素は５つですから、i の値は 0～4 となります。

for 文の中では、if 文を使って該当のラジオボタンがチェックされているか確認しています。チェックされているなら正しい答えを選択していることになるので、 変数 seikai を 1 追加します。

最後に、変数 seikai に 20 を掛けた点数を alert で表示しています。5 問全てに正解すると、100 点になります。

このようにして三択クイズを作ることができますが、HTML ソースを覗かれると答えがバレてしまいます。 関数は外部ファイルに記入して、少しでも目につきにくくしておきましょう。 （それでも分かる人には覗かれますが。。。）

次のページでは、セレクトボックスの使い方について解説します。

セレクトボックスの操作

このページでは、セレクトボックスを JavaScript で扱う方法について解説します。 現在選択されている項目を取得したり、 セレクトメニュー項目を設定する方法について見てみます。

セレクトボックスを扱うサンプルスクリプト

では、以下のコードを HTML のボディ内に記入してみてください。 フォーム部分はコピー&ペーストでも構いませんが、 JavaScript 部分は手書きでチャレンジしてもらえたら理解も深まると思います。

```
<form name="selbox">
<p>好きなプロ野球リーグは？</p>
<select name="league" onchange="teamSet()">
<option value="">*リーグ選択</option>
<option value="">セ・リーグ</option>
<option value="">パ・リーグ</option>
</select>

<p>好きなチームは？</p>
<select name="team">
<option value="">*チーム選択</option>
<option value=""></option>
<option value=""></option>
<option value=""></option>
<option value=""></option>
<option value=""></option>
<option value=""></option>
</select>
</form>


<script>

//セ・リーグのチームの配列
var s_league=new Array(
"","中日","ヤクルト","巨人","阪神","横浜","広島"
);
```

```
//パ・リーグのチームの配列
var p_league=new Array(
"","ロッテ","西武","オリックス","ソフトバンク","楽天","日本ハム"
);


function teamSet(){

 //オプションタグを連続して書き換える
 for ( i=1; i<7; i++ ){

  //選択したリーグによって分岐
  switch (document.selbox.league.selectedIndex){
   case 0: document.selbox.team.options[i].text="";break;
   case 1: document.selbox.team.options[i].text=s_league[i];break;
   case 2: document.selbox.team.options[i].text=p_league[i];break;
  }
 }

 //チーム名のセレクトボックスの選択番号を０にする
 document.selbox.team.selectedIndex=0;
}

</script>
```

「チーム選択」のセレクトボックスは最初は項目が何も表示されていません。 しかし「リーグ選択」セレクトボックスでどちらかのリーグを選択すると、 チーム名が表示されるようになります。

サンプルスクリプトの解説

では上記のサンプルスクリプトを見ていきましょう。フォーム部分では、例によって name 属性を付けています。 フォームは「selbox」、最初のセレクトボックスは「league」、 ２番目のセレクトボックスを「team」にしました。

セレクトボックスが変更された時に何か JavaScript を動かすときは、 onchange イベントを用います。上記のソースでは最初のセレクトボックスを変更したときに、関数 teamSet() が実行されます。

続いて JavaScript 部分を見ていきます。 まずは各リーグのチーム名を入れる配列「s_league」,「p_league」を作成し、 チーム名を登録しています。 各配列の最初に空文

字をセットしていますが、単なる順番合わせです （最初の option タグ「*チーム選択」は今回書き換えないので、それに対応した s_league[0]，p_league[0]も参照しません)。
そして、関数 teamSet()を記述します。 その中で for 文を使って、 team セレクトボックスの２つ目以降の option タグのテキストを書き換えます。 チームは６つあるので、6 回繰り返します。 i の初期値が「0」ではなく「1」なのは、2 番目の option タグから書き換えるからです。
for 文の中で switch 文を使い、league セレクトボックスのどれが選択されているかによって処理を分岐します。 分岐条件はセレクトボックスの選択項目番号です。 selectedIndex を使うと取得できます。
document.form 名.select 名.selectedIndex
選択番号を取得します
「セ・リーグ」を選ぶと 1,「パ・リーグ」を選ぶと 2、 「*リーグ選択」を選ぶと 0 となります。あとは case を使って分岐し、それぞれの処理を記述します。
option タグのテキストを書き換えるには、options[].text を使います。 options[]は配列になっているので、セレクトボックスの最初の項目は options[0]となります。 for 文を 1 から始めたのも、optoions[0]の「*チーム選択」は書き換えないからです。
document.form 名.select 名.options[].text
option タグのテキストを取得します。
「セ・リーグ」を選んだ時は、配列 s_league の各項目を option タグのテキストにセットしていきます。 チーム名配列の最初に空項目を入れたのは、配列 s_league と配列 options[]の番号を合わせるためです。
「パ・リーグ」を選んだ時は、配列 p_league の各項目をセットします。 「*リーグ選択」を選んだ時は、空文字をセットして option タグのテキストを消しています。 case 文の最後は break を忘れずに入れます。
最後に、team セレクトボックスの選択番号を 0 にして、「*チーム選択」を表示するようにしています。 selectedIndex は選択番号を取得するだけでなく、 代入することによって目的の項目を選択させることもできます。
禁止状態
いきなりですが、下のボタンを押してみてください。
ボタンを押すと、淡い色に変化してもう押せなくなったと思います。 ２度押しされると困るような時に使うことができます。 このようにフォームの要素を使用できない状態にする方法について調べてみることにしましょう。
禁止状態のサンプルスクリプト
下の例は、名前と感想を記入してもらったらボタンが押せるようになるスクリプトです。
では、以下のソースを HTML の body 内に記入してみましょう。

```
<form name="send">
```

```
お名前：<input type="text" value="" onchange="wupBtn()"><br>
感　想：<textarea rows="2" cols="20" onchange="wupBtn()">
</textarea><br>
<input type="button" value="送信" disabled>
</form>


<script>

function wupBtn(){

  //名前と感想の欄のテキストを変数に代入する
  var namae = document.send.elements[0].value;
  var kanso = document.send.elements[1].value;

  //名前若しくは感想欄のどちらかが空かチェック
  if ( ( namae == "" ) || ( kanso == "" ) )
  {
    //どちらか空であれば、ボタンを押せなくする
    document.send.elements[2].disabled = true;
  }else{
    //両方とも書き込まれていたら、ボタンを押せるようにする
    document.send.elements[2].disabled = false;
  }
}

</script>
```

以下はサンプルです。ちなみにボタンを押しても何も起こりません。

disabled の解説

では上記スクリプトについて見ていきましょう。まずフォーム名は「send」にしています。テキストボックスとテキストエリアには onchange イベントを記入し、 変更が加えられたなら関数 wupBtn()を呼び出すようにしています。 ボタンは最初押せないように disabled を記入しておきます。

次に JavaScript 部分を見ていきましょう。 最初に関数 wupBtn()を作成します（wup は wake up のつもり）。

関数の中で始めに、変数 namae と変数 kanso を宣言し、 テキストボックスとテキストエリアの文字列を代入しています。 テキストエリアの文字列も value で取得できます。
そして if 文で、名前欄若しくは感想欄のどちらかが空かチェックします。 中央の「||」は「or」の意味でした。
どちらかが空であればボタンを押せなくし、そうでなければボタンを押せるようにします。
禁止状態の ON/OFF は disabled を使います。
document.form 名.elements[].disabled = true/false
true なら禁止状態に、false なら変更可能にする
因みにこのページの最初に出てきた「2 度押しできないボタン」は、 ボタンの onclick イベントで自身を禁止状態にすれば実現できます。
フォームの小技
フォーム編の最後としてちょっとした小技を幾つか紹介します。 フォーカスを移動したり、テキストを選択状態にしてみます。
フォーカスを操作する
最初にフォーカスを合わせたり外したりしてみます。 これは、最初に入力してもらいたい枠にフォーカスを合わせておく場合などに使用します。

```
<form name="fcset">
<input type="text" value="text1">
<input type="button" value="合わせる" onclick="fc(0)">
<input type="button" value="外す" onclick="bl(0)"><br>

<input type="text" value="text2">
<input type="button" value="合わせる" onclick="fc(3)">
<input type="button" value="外す" onclick="bl(3)"><br>
</form>

<script>

//フォーカスを合わせる関数
function fc(num){
   document.fcset.elements[num].focus();
}

//フォーカスを外す関数
function bl(num){
   document.fcset.elements[num].blur();
```

```
}
</script>
```

フォームの名前は「fcset」にしました。 二つのテキストボックスと、それぞれにフォーカスを合わせる/外すためのボタンを設置しています。

フォーカスを合わせたり、外したりする JavaScript の命令文は以下のようになります。

```
document.form 名.elements[].focus()
```

フォーカスを合わせる

```
document.form 名.elements[].blur()
```

フォーカスを外す

上記のスクリプトでは、それぞれの関数に引数「num」を設けて、 フォーカスを ON/OFF するテキストボックスを選択できるようにしました。 最初のテキストボックスは elements[]の番号が０、次のテキストボックスは３になるので、 関数を呼び出す時にそれぞれの数字をカッコ内に入れています。

テキストを選択する

テキストボックスやテキストエリアの文字列を全て選択状態にして、 コピーしてもらい易くする方法についてみてみます。 上記の関数 fc()に下記のように一行追加してみてください。

```
function fc(num){
   document.fcset.elements[num].focus();
   document.fcset.elements[num].select();
}
```

今度はフォーカスが合った時に自動的にテキスト選択状態になります。 （ブラウザによっては focus()だけでも選択状態になりますが、こちらの方が確実です)。 テキストを選択状態にする命令文は以下の通りです。

```
document.form 名.elements[].select()
```

テキストを選択状態にする

その他の操作

その他にできる操作について簡単に掲載しておきます。

```
document.form 名.elements[].click()
```

自動的にクリックする

```
document.form 名.reset()
```

初期化する（リセットボタンと同じ役目）

```
document.form 名.submit()
```

送信する（submit ボタンの同じ役目）

初期化と送信の場合は elements[]は付きません。 submit()はブラウザやユーザーの設定に

より正しく動作しないことが多いので、 使わない方が無難です。