

confirm で正誤クイズを作る

さて、ここからダイアログを使ったプログラムを組む練習をしてみましょう。すでに [alert](#) は基礎編で出てきたので割愛します。ここでは **confirm** というダイアログを使ってみます。これはユーザーに選択を促すダイアログです。

上図のようなダイアログがでます。処理の前に確認を取ったりする時に使います。

confirm の使い方

まずは **confirm** を使い方を解説しておきます。以下のように HTML の body 内に書いて下さい。

```
<script>
```

```
var flg; //真偽を入れる変数の宣言
```

```
flg = confirm("背景の色を変えてもいいですか?"); //真偽の代入
```

```
if (flg == true) document.bgColor = "green"; //真なら背景色変更
```

```
</script>
```

解説しますと、まずユーザーの選択（OKかキャンセルか）を入れる **flg** という変数を宣言しています。この変数は **boolean** 型という型にしようと思います。この型は真（**true**）か偽（**false**）のどちらかしか入らないものです。

次に **confirm()** が出てきます。この括弧内の文字がダイアログで表示されます。文字列はクォーテーションで囲むのが約束ですね。そしてユーザーがOKを押すと、変数 **flg** に真（**true**）が、キャンセルを押すと偽（**false**）が代入されます。

confirm(質問文)

確認ダイアログ（OKかキャンセルを選択）を出します。

OKなら **true**、キャンセルなら **false** となります。

最後に [if 文](#) を使って変数 **flg** を判断します。 **true** であれば、背景を変更します。上述の **document.bgColor** でページ背景色を変更できます。

正誤クイズを作ってみる

では、正誤クイズを作ってみたいと思います。5問連続で出してみましょう。HTML のヘッダーに以下の関数を記入してみて下さい。

```
//問題を配列に入れる
```

```
var q = new Array(
```

```
"水を電気分解すると水素と酸素が出る。",
```

```

"地球上で最も多い元素は炭素である。",
"金の元素記号は Gd である。",
"血液は骨の内部で作られる。",
"動物細胞に細胞壁は無い。"
);

//答え（正誤）を配列に入れる
var ans = new Array(true,false,false,true,true);

function quiz(){
    var ten = 0; //点数を入れる変数

    // 5 問連続出題
    for(var i = 0 ; i <= 4 ; i++){
        if( confirm(q[i]) == ans[i] ) ten = ten + 20;
    }

    //採点結果発表
    alert("あなたは" + ten + "点でした！");
}

```

あとは、body 内で関数 quiz()を呼び出します。

では解説してみます。 まずは配列 q に問題を、配列 ans に解答（true か false か）を入れています。 [配列](#)を使うのは、for 文と共に連続出題するためです。

そして、関数 quiz()の宣言です。 関数内でまず点数を入れる変数の宣言をしています。 変数 ten を宣言し、初期値 0 を代入します。

次に [for 文](#)を使って 5 連続で問題を出します。（配列を使っているので i の取る値は 0 から始めます）。 そして confirm のカッコ内には q[i]を指定し、 問題をダイアログで表示します。

今回は cnfirm の選択結果を変数に代入せず、ダイレクトに if 文で真偽を判断します。 判断値は ans[i]となっています。 配列 ans は配列 q の問題の答えに対応していなければいけません。 正誤が一致すれば、変数 ten の値に 20 を加えます。

最後に alert で結果を出力しています。

あとは ten を判断して 100 点ならパスワードを出したり、 0 点なら罰ゲームをさせたりすることもできます。 いろいろ工夫してみましょう。

prompt で名前を表示する

あるサイトを訪れた時、インプットボックスが現れて名前の入力求められたという経験はありませんか？ 入力すると自分の名前がサイト内に表示されたりしますね。ここではそんなインプットボックスを出す方法について解説してみます。

インプットボックスはうちのサイトの[クロスワード](#)とか、パズル全般のパスワードを表示するのに用いています。色々使い道があるので、是非覚えて下さい。

インプットボックスを出す

インプットボックスを出すには、**prompt** という命令を使います。 **prompt** を使って名前を入力してもらい、 入力文字をページ上に表示するサンプルを作ってみることにしましょう。

HTML の body 内に以下のように記入して下さい。

```
<script>
```

```
var str; //入力文字を入れる変数
```

```
//インプットボックスの表示&入力文字を str に代入
```

```
str = prompt("お名前を教えてください。","");
```

```
//str が空の場合の処理
```

```
if (str == "") str = "名無しの権兵衛";
```

```
//名前を表示する
```

```
document.write("<center>");
```

```
document.write("ようこそ、<b>" + str + "</b>さん！");
```

```
document.write("</center>");
```

```
</script>
```

上から順に解説してみましょう。 まず変数 **str** を宣言します。 これはビジターが入力してくれた文字を入れるための変数です。

次に **prompt()** でインプットボックスを出します。 括弧内は2つの区分があります。 最初の部分は質問文を記入し、後ろの部分は入力枠に表示する文字を指定します。 今回は入力枠に何も表示する必要が無いので、空にしています。 文字列ですからクォーテーションで囲み、2つの区分の間は半角コンマで区切ります。

```
変数 = prompt("質問文","最初に表示する文字");
```

```
インプットボックスを出します。
```

```
カッコ内の最初の項目は質問文、2番目の項目は初期表示文字を指定します。
```

```
ユーザーの入力した文字列は、変数に代入されます。
```

次に、ビジターが何も入力せずにOKを押した場合の処理を書きます。 [if文](#) で **str** を調べ、

空であれば「名無しの権兵衛」を代わりに代入します。

最後に [document.write](#) で歓迎文を書き出します。上のサンプルでは3回に分けて書いていますが、これは画面の横幅の関係で1行で書けなかったためです。別に1つにまとめて書いてくださって構いません。

キャンセルを押された時の処理

さて、上記のソースのままだと、インプットボックス上でキャンセルを押された場合に不具合が出ます。試しにやってみると分かりますが、**null** と表示されると思います。

この **null** というのは、データが無い状態を表します。OKを押した場合は、空ですが一応文字列データ""が代入されます。しかしキャンセルを押された時はそのデータすら無い状態なのです。

ということで、上記ソースの if 文の後に、もう1つチェック文を付け加えましょう。

```
if (str == "") str = "名無しの権兵衛";
```

```
if (str == null) str = "名無しの権兵衛";
```

以下のようにまとめて書くことも可能です。'||' は or (どちらか) という意味です。

```
if ( (str == "") || (str == null) ) str = "名無しの権兵衛";
```

これでキャンセルを押されても大丈夫です。prompt でキャンセルを押された場合は **null** という状態になることを覚えておきましょう。

パスワード入力で秘密のページへ

ここでは [prompt](#) を用いてパスワードを入力してもらい、秘密のページへジャンプさせる方法について解説してみます。と言っても、JavaScript で作るものはお遊び程度なので、ちょっと知識のある人であれば簡単に見破られてしまいます。あまり重要なページには用いないようにしましょう。

パスワード入力による移動

では、「abcde」というパスワードを入力すると隠れページにジャンプするソースを書いてみましょう。

```
<script>
```

```
var pw;
```

```
pw = prompt("パスワードを入れて下さい。","");
```

```
if (pw == "abcde"){
```

```
    location.href = "kakure.html";
```

```
}else{
```

```
    alert("パスワードが違います!");
```

```
}
```

```
</script>
```

ソースを見ていきましょう。最初に変数 `pw` を宣言しています。そして [prompt](#) でインプットダイアログを出し、パスワードを入力してもらいます。入力された文字は、変数 `pw` に代入されます。

次に [if 文](#) で、`pw` がパスワード「abcde」と等しいかチェックします。等しい場合は [location.href](#) で隠れページに移動し、等しくない場合は [アラート](#) を出します。

しかしご覧の通り、ソースを見るとパスワードも隠れページの URL もバレバレですね。 [外部ファイル](#) に記入すれば少しはマシですが、ちょっと知識のある人であれば外部ファイルを覗けますからあまり意味がありません。では、もう少しバレ難い方法について調べてみましょう。

バレ難いパスワード移動方法

ここで紹介する方法は、私が JavaScript を勉強した「[イヌでもわかる JavaScript](#)」で紹介されていた方法です。なかなか考え抜かれた方法だと思います。

```
<script>
```

```
var pw;
```

```
pw = prompt("パスワードを入力して下さい。","");
```

```
if (pw != "")
```

```
location.href = pw+".html";
```

```
</script>
```

`prompt` でパスワードを入力してもらうまでは同じです。違うのは飛び先指定の所です。「パスワードの文字+拡張子」がそのまま飛び先 URL になっていますね。これだとソースを見てもパスワードがバレることはありません。当然隠れページはパスワードと同じファイル名になります。（パスワードが「abcde」なら、隠れページは `abcde.html` とします）。

まあこれで完璧なように見えますが、実際はフリーの解析ソフトなどを使っても隠れページが見つかったりすることがあります。遊び程度に用いるのが無難でしょう。