

サブウィンドウを表示する

ここでは、**JavaScript** でサブウィンドウを表示する方法について解説します。 **HTML** でも別ウィンドウは出せますが、 **JavaScript** を利用するとウィンドウの位置や大きさを指定できます。 何かと使い道の多いスクリプトですので、是非マスターして下さい。

普通にサブウィンドウを表示する

まずは普通にサブウィンドウを出してみたいと思います。 **HTML** の **target** 属性と同じ働きをします。 ヘッダーに以下の[関数](#)を記入してみてください。

```
<script>
```

```
function winOpen(){
```

```
window.open("../index.html","sub");
```

```
}
```

```
</script>
```

そして、BODY 内で関数を呼び出します。(下記例のように)

```
<form>
```

```
<input type="button" value="別窓" onclick="winOpen()">
```

```
</form>
```

上記ボタンを押すと、1 つ上の階層の **index.html**、ここでは **JavaScript** 入門の **TOP** ページが表示されます。 ではサブウィンドウを出すスクリプトを解説してみることになります。

window.open の解説

上記スクリプトに出てくる **window.open()** というのが、 サブウィンドウを出す命令です。 カッコ内には、最初に表示させるページの **URL** を、 次はウィンドウ名を指定します。 どちらも文字列ですから、クォーテーションで囲います。

```
window.open("URL","ウィンドウ名")
```

指定したウィンドウ名でサブウィンドウを表示させます

ウィンドウ名は、**HTML** のターゲット名と同じです。 同じウィンドウ名を指定すると、そのウィンドウに指定したページが表示されます。 ウィンドウ名を空にすると、**target="_blank"**と同じことになります (下記)。

```
window.open("../index.html","");
```

また、サブウィンドウを開く関数は汎用性を持たせるため、普通は[引数](#)を用いて以下のようになります。

```
<script>
```

```
function winOpen(jpURL){
```

```
    window.open(jpURL,"");
```

```
}
```

```
</script>
```

これで関数を呼び出す時、引数に URL を指定すればどのページでも表示できます。よく使う関数は[外部ファイル](#)にしておくと便利です。

次ページでは、サブウィンドウの大きさと表示位置を指定する方法を解説します。

サブウィンドウのサイズと位置

[前のページ](#)でサブウィンドウを表示する方法を見ましたが、このページでは新しいウィンドウのサイズと位置を指定する方法について解説します。

別ウィンドウのサイズと位置を指定する

以下の関数を、HTML のヘッダーに記入してみてください。

```
<script>
```

```
function subwin(jpURL){
```

```
    window.open(jpURL,"","width=640,height=480,top=0,left=0");
```

```
}
```

```
</script>
```

次に、body 内にフォームタグとボタンを記入し、onclick イベントに以下のように記して下さい。

```
<form>
```

```
<input type="button" value="TOP へ"
```

```
    onclick="subwin('http://pori2.net/')">
```

```
</form>
```

下のボタンはそのサンプルです。当サイトのトップページ (<http://pori2.net/>) が小さめのサブウィンドウ (640×480 ピクセル) に表示されます。

window.open()の位置とサイズの指定方法

では、window.open()のサイズと位置を指定するオプションパラメーターの解説をします。window.open のカッコ内は、以下のような順番で設定していきます。

`window.open("URL","ウィンドウ名","オプション")`

オプションを指定したサブウィンドウを表示します。

最初の「URL」と2番目の「ウィンドウ名」は[前のページ](#)で説明しました。このページでは3番目の「オプション」のうち、サイズと位置を指定するパラメーターについて考えます。そのパラメーターとは、以下のものです。

width

ウィンドウの横幅を指定します。サンプルでは 640 ピクセル。

height

ウィンドウの縦幅を指定します。サンプルでは 480 ピクセル。

top

デスクトップ上端からの距離です。サンプルでは 0 ピクセル。

left

デスクトップ左端からの距離です。サンプルでは 0 ピクセル。

オプション全体は、クォーテーションで括り、各パラメーターはコンマで区切ります。各オプションの順番は特に決まりはありません。

ただ、このサブウィンドウにはスクロールバーが表示されていないかも知れません。これはブラウザによって若干違うのですが、Internet Explorer 8 と FireFox 3.5 では表示されません。つまりスクロールできない状態にあります。(Windows 版の Safari4.0 では表示されます)。

このスクロールバーや、ツールバーの表示の ON/OFF も、オプションパラメーターで設定できます。次のページでは、そうした設定について解説してみます。

window.open()のオプション

前のページでは、別ウィンドウのサイズと位置を指定する方法について見ました。このページでは、サブウィンドウのスクロールバーやメニューバー等の ON/OFF の設定について解説します (しかしブラウザやそのバージョンによって挙動は様々であり、うまく動かないことも多々あります。その点はしっかり銘記しておきましょう)。

オプション変更関数の作成

まずは以下の[関数](#)をヘッダーに記入して下さい。

```
<script>
```

```
function subWin(option){  
    window.open("test.html","sub","width=640,height=480,top=0,left=0"  
        +","+option);  
}
```

関数 subWin()の[引数](#)に option を指定しています。そして関数の中で window.open()を呼び出して、サブウィンドウに表示するページを「test.html」、ウィンドウ名を「sub」、サイズは 640×480 ピクセル、位置は左上にしています。（test.html の部分は自由に変更して下さい）。

またオプション部分はクォーテーションで括っていることから文字列とみなされることが分かります。ということで、上記のように「+」を使って繋げることが可能です。今回は引数 option にいろいろなオプション属性を指定して調べてみたいと思います。

スクロールバー他の ON/OFF

では BODY 内に以下のタグを記入してください。

```
<form>  
<input type="button" value="スクロール ON"  
    onclick="subWin('scrollbars=yes')">  
<input type="button" value="スクロール OFF"  
    onclick="subWin('scrollbars=no')"><br>  
</form>
```

上記のように、scrollbars がスクロールバーに関するオプションパラメーターです。yes なら表示、no なら非表示になります。他のオプションパラメーターと共に、以下に載せておきます。

フォームの始まり

scrollbars=yes/no

スクロールバーの表示、非表示を指定します。

toolbar=yes/no

ツールバーの表示、非表示を指定します。

menubar=yes/no

メニューバーの表示・非表示を指定します。

status=yes/no

ステータスバーの表示・非表示を指定します。

directories=yes/no

ディレクトリバーの表示・非表示を指定します。

`resizable=yes/no`

ウィンドウサイズ変更の許可・不可を指定します。

さて、サブウィンドウを表示させてみて分かると思いますが、指定通りにならないものもあります。実はこのオプション指定、ブラウザによって挙動はまちまちなのです。このことを頭の片隅に必ず置いておいて下さい。でないと思わぬ失敗をするかも知れません。

※最近のブラウザでは殆どの場合、指定は無視されるようです。

オプションパラメーターを複数記入する時

最後に、これらのパラメーターは複数指定も可能だということを付け加えておきます。下記のように記入することができます。

```
window.open("test.html","sub","width=640,height=480,top=0,left=0"+  
    ",scrollbars=no,toolbar=no,menubar=no,status=no"+  
    ",directories=no,resizable=no");
```

オプションパラメーターを沢山記入する場合はとても長くなるので、上の例では3行に分けて書いています。この時、行を変える場合は必ずクォーテーションを一度閉じて、「+」を用いて繋げるようにしましょう。クォーテーション内で改行すると即エラーになります（下記の例ではエラーになります）。

```
window.open("test.html","sub","width=640,height=480,top=0,left=0  
    ,scrollbars=no,toolbar=no,menubar=no,status=no  
    ,directories=no,resizable=no");
```

次のページでは、サブウィンドウを画面中央に表示する方法を考えてみましょう。

別ウィンドウを画面中央に表示する

ここでは、サブウィンドウを大体画面の中央に表示する方法について考えます。画面サイズはそれぞれのPCで違うので、`window.open()`のオプションパラメーター、[left](#) と [top](#) の値を決めるのに一工夫必要となります。

別窓を画面中央に表示する関数

では640×480のウィンドウを画面中央に表示してみます。HTMLのヘッダーに以下のように記入してみてください。

```
<script>
```

```
function winCenter(){
```

```
//別窓の左と上の余白を求める
```

```
var w = ( screen.width-640 ) / 2;  
var h = ( screen.height-480 ) / 2;
```

```
//オプションパラメーターleft と top に余白の値を指定  
window.open("../index.html","", "width=640,height=480"  
    +",left="+w+",top="+h);  
}
```

```
</script>
```

そして BODY 内にフォームとボタンを設置し、[関数](#) winCenter() を呼び出します。

```
<form>
```

```
<input type="button" value="中央表示" onclick="winCenter0">
```

```
</form>
```

上記の例では、この JavaScript 入門の INDEX ページがサブウィンドウに表示されます。ウィンドウが大体画面の中央に表示されたでしょうか？

画面サイズから左と上の余白を導き出す

では、上記のスクリプトについて見ていきましょう。 まず「 w 」と「 h 」という二つの[変数](#)を宣言しました（width と height の頭文字のつもり）。 その右辺にある screen.width が画面の横幅、 screen.height が画面の縦幅を取得する命令です。

screen.width

画面の横幅を取得します

screen.height

画面の縦幅を取得します

画面横幅から別窓横幅を引くと、横の余白が求められます。 それを 2 で割っていますが、余白を半分にして別窓の右と左に均等に振り分けています。（縦に関しても同じように考えます）。

(画面横幅－別窓横幅) ÷ 2 = 左右の余白

こうして取得した縦横の余白を、window.open() のオプションパラメーター、 left と top に指定すれば OK です。 パラメーターと変数を繋ぐときは、一度クォーテーションを閉じて「 + 」を使って結びます。

ただ、実際にはデスクトップにタスクバーがありますし、 別窓もタイトルバーがあるので厳密な意味で中央には表示されません。 このページで大体中央にと述べているのは、そういう意味です。

次のページでは、サブウィンドウを開いた後に移動する方法について解説します。

サブウィンドウの移動

これまでは別ウィンドウを「開く段階」で、[表示する位置を指定](#)していました。このページでは別ウィンドウを「開いた後」に、 任意の場所に移動する方法について解説します。

別窓を開く関数

まずは移動する別窓を表示させる必要があるので、その[関数](#)を作成します。 320×240 ピクセルの別窓を、画面上の座標（100,50）に表示させてみます。 HTML のヘッダーに以下の関数を記入して下さい。

```
<script>

function winOpen()
{
    window.open("movewin.html","subwin",
        "width=320,height=240,left=100,top=50");
}
</script>
```

そして body 内でこの関数を呼び出すボタンを作ります。

```
<form>
<input type="button" value="サンプル" onclick="winOpen()">
</form>
```

これで準備完了です。

別ウィンドウを移動させる

次に別窓で表示するページを作ります。前述の関数 winOpen()では、movewin.html を開くように指定していますので、 そのファイル名で HTML ファイルを作成して下さい。そして movewin.html のヘッダーに以下のように記入します。

```
<script>

//ウィンドウを相対的に移動
function move1(yoko,tate){

    moveBy(yoko,tate);

}
```

```
//ウィンドウを座標指定で移動
function move2(yoko,tate){
```

```
    moveTo(yoko,tate);
}
}
```

```
</script>
```

関数 `move1()` は、ウィンドウを相対的に移動させ、関数 `move2()` は、ウィンドウを指定した座標に移動させます。相対指定の場合は `moveBy()` を、絶対位置指定（座標指定）の場合は `moveTo()` を使います。「B」と「T」が大文字なので気を付けて下さい。

`moveBy()` (横の移動距離, 縦の移動距離)

ウィンドウを指定した数値分移動します

`moveTo()` (横の座標, 縦の座標)

ウィンドウを指定した座標に移動します

各関数には [引数](#) が2つありますが、それぞれ横・縦の移動距離 or 座標を柔軟に指定できるようにしています。

最後に、`movewin.html` の `body` 内に以下のボタンを設置してみましょう。

```
<form>
<input type="button" value="右へ" onclick="move1(10,0)"><br>
<input type="button" value="左へ" onclick="move1(-10,0)"><br>
<input type="button" value="上へ" onclick="move1(0,-10)"><br>
<input type="button" value="下へ" onclick="move1(0,10)"><br>

<input type="button" value="元の位置" onclick="move2(100,50)">

</form>
```

方向を示す各ボタンを押すと、目的の方向へ 10px ずつ移動します。元の位置ボタンを押すと、最初に別窓を開いた位置(100,50) に移動します。

この `moveBy()` や `moveTo()` は別窓だけでなく、メインウィンドウでも使えます（以下にサンプル）。しかしブラウザの表示位置を勝手に変えられると怒る人もいるので、別ウィンドウのみで使用しましょう。

次のページでは、別窓を開いた後にリサイズする方法について解説します。

サブウィンドウのリサイズ

これまでのところでは、サブウィンドウを「開く段階」でウィンドウサイズを指定していましたが。このページでは、サブウィンドウを「開いた後」にリサイズする方法について考えます。

別窓を開く関数

まずはリサイズさせる別ウィンドウを開く[関数](#)を作ります。HTML のヘッダーに以下のように記入して下さい。 320×240 ピクセルのウィンドウを画面左上に表示します。

```
<script>
function winOpen()
{
    window.open("winresize.html","subwin",
        "width=320,height=240,left=0,top=0");
}
</script>
```

そして BODY 内で winOpen()関数を呼び出します。たまには [A タグ](#) を使って呼び出してみます。

```
<a href="javascript:winOpen()">サンプル</a>
```

これで準備完了です。

別ウィンドウをリサイズする

次に別窓で表示するページを作ります。上の関数 WinOpen()では、winresize.html を開くようにしていますので、そのファイル名で HTML ファイルを作成してください。そしてヘッダーに以下のように記入します。

```
<script>
```

```
//ウィンドウを相対的にリサイズ
```

```
function reSize1(yoko,tate){
```

```
    resizeBy(yoko,tate);
```

```
}
```

```
//ウィンドウを指定サイズに変更
```

```
function reSize2(yoko,tate){
```

```
resizeTo(yoko,tate);
```

```
}
```

```
</script>
```

関数 `reSize1()` は、ウィンドウを相対的にリサイズさせ、関数 `reSize2()` は、ウィンドウを指定サイズにリサイズします。相対指定の場合は `resizeBy()` を、絶対指定（指定サイズ）の場合は `resizeTo()` を使います。前頁の `moveBy()`, `moveTo()` の時と同じく、「B」と「T」が大文字なので気を付けて下さい。

resizeBy(横の変更幅,縦の変更幅)

ウィンドウを指定した数値分伸縮させます

resizeTo(横のサイズ,縦のサイズ)

ウィンドウを指定サイズに変更します

各関数には引数が2つありますが、それぞれ横・縦の変更幅 or サイズを柔軟に指定できるようにしています。

最後に、`winresize.html` の BODY 内で関数 `reSize1()`、関数 `reSize2()` を呼び出します。ここでも A タグを使ってみます。

```
<a href="javascript:reSize1(10,0)">横伸</a><br>
```

```
<a href="javascript:reSize1(0,10)">縦伸</a><br>
```

```
<a href="javascript:reSize1(-10,0)">横縮</a><br>
```

```
<a href="javascript:reSize1(0,-10)">縦縮</a><br>
```

```
<br>
```

```
<a href="javascript:reSize2(320,240)">指定サイズ</a><br>
```

縦横の伸縮リンクをクリックすると、各方向に 10 ピクセルずつ伸縮します。指定サイズをクリックすると、 320×240 のサイズになります。

この指定サイズは元のウィンドウと同じ 320×240 ピクセルに設定していますが、実際にクリックしてみるとさらに縮まることが分かります。つまり、`window.open()` で指定した場合、ウィンドウの表示部分（BODY 部分）のサイズを指定でき、`resizeTo()` で指定した場合はウィンドウそのもののサイズを指定するようです。

このリサイズ機能も前項の `moveBy()` や `moveTo()` と同じくメインウィンドウでも使えます。ただしメインウィンドウを勝手にリサイズされると怒る人もいるので、絶対に止めましょう！

次のページでは、親ウィンドウの情報をサブウィンドウに伝える方法について解説します。

オープン元参照&ウィンドウを閉じる

このページでは、サブウィンドウからオープン元である親ウィンドウの情報を取得する方法について解説します。 またウィンドウ編の最後として、サブウィンドウを閉じる方法についても解説します。

オープン元のページ

まずはオープン元となる HTML ファイルに、 サブウィンドウを開く[関数](#)と、サブウィンドウに渡す情報を記入してみましょう。 HTML のヘッダーに以下のように記入してみてください。

```
function winOpen(){
```

```
    window.open("subwin.html","sub");
```

```
}
```

そして body 内に、以下のフォームを記入します。 サブウィンドウからデータを受け取るために、name 属性を付けます。

```
<form name="sample">
```

```
<input type="button" value="サンプル" onclick="WinOpen()">
```

```
<input type="hidden" name="suji" value="12345">
```

```
<input type="hidden" name="moji" value="あいうえお">
```

```
</form>
```

これでオープン元の HTML ファイルは準備できました。

オープン元を参照する

今度は、サブウィンドウからオープン元のウィンドウを参照してみます。 上述の関数 winOpen() では subwin.html を開くようになっていしますので、 そのファイル名で HTML ファイルを作成して下さい。 そして subwin.html の body 内に以下のように記入します。

```
<script>
```

```
//オープン元を参照して変数に代入する
```

```
var num=window.opener.document.sample.suji.value;
```

```
var str=window.opener.document.sample.moji.value;
```

```
document.write(num+"<br>");
```

```
document.write(str+"<br>");
```

```
</script>
```

サンプルページを開いてみると、オープン元のフォームにある2つの隠し属性に記入した

「12345」と「あいうえお」が表示されると思います。

上記のコードにあるように、`window.opener` を使ってオープン元を参照します。続く部分に、フォームの隠し属性のデータを取得するコードを書きます。隠し属性のデータを取得する方法は以下の通りです。

```
window.opener.document.form 名.input 名.value
```

オープン元のフォームの情報を取得する

`input` タグは別に隠し属性だけでなく、テキストボックスや[セレクトボックス](#)でも使えます。

ウィンドウを閉じる

サブウィンドウを閉じる方法について触れておきます。今回作った `subwin.html` の `body` 内に、以下のフォームを付け足して試してみてください。

```
<form>
<input type="button" value="閉じる" onclick="window.close()">
</form>
```

ウィンドウを閉じるには、`window.close()`を使います。ブラウザのウィンドウが1つしか開いていない場合は、機能しません。

```
window.close()
```

ウィンドウを閉じます

スクロール

ここでは、ページを上下にスクロールさせる方法について取り上げます。特定の位置にジャンプしてもらいたい時や、タイマーと組み合わせてスタッフロールを作ったりすることができます。

ページをスクロールさせる

では、以下のスクリプトを `HTML` の `BODY` 内にコピー&ペーストしてみてください。改行タグを沢山入れて、スクロールバーを表示させています。

```
ページ先頭<br>
<br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br>
<form>
<input type="button" value="上へ" onclick="goTop()">
<input type="button" value="200px 下へ" onclick="down()">
</form>
<br><br><br><br><br><br><br><br><br>
```

```
<br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br>
```

続いて HTML のヘッダーに以下の 2 つの[関数](#)を記入して下さい。

```
<script>
```

```
//ページの先頭に戻る
```

```
function goTop()
```

```
{
```

```
    scrollTo(0,0);
```

```
}
```

```
//200 ピクセル下に移動する
```

```
function down()
```

```
{
```

```
    scrollBy(0,200);
```

```
}
```

```
</script>
```

下のサンプルをご覧ください（ボタンが見える所までページを下にスクロールしてください）。
「上へ」ボタンを押すと、ページ先頭に戻ります。「200px 下へ」を押すと、ページが少し下にスクロールします。

ページスクロールのスク립ト解説

では上記のソースについて見てみましょう。 最初の関数 `goTop()` の中では `scrollTo()` が使われています。 これはページの指定位置にスクロールさせる命令文です。 括弧内は最初が X 座標（横位置）、2 目が Y 座標（縦位置）で、 横長のページでは横にもスクロールさせることができます。

2 目の関数 `down()` の中では、`scrollBy()` を使用しています。 これは現在の位置から相対的にスクロールさせる命令文です。 括弧内はやはり最初が横位置、2 番目が縦位置となります。 上に移動させたい場合は、2 番目にマイナスの数値を記入すれば OK です。

`scrollTo(X 座標,Y 座標)`

ページ内の指定位置にスクロールします。

`scrollBy(X 方向の移動量,Y 方向の移動量)`

現在位置から相対的なポイントにスクロールします。