

HTML TIPS & TRICKS

第 35 回

誰よりも早く 最新のHTMLを使ってみよう

藤井幸孝 / 大内 勇

ここ何か月かIE 5の最新機能を使ったTIPSを紹介してきたが、今月は一休みしてIE 3やナビゲーター3でも動作するスクリプトや、IE 4の知られていないプロパティなどをお見せしよう。12月号のパズルの解答を見ればわかるように、古いテクニックでも組み合わせれば大きな効果を出すことができる。それでは今月もさまざまなテクニックをお届けしよう。パズルも面白いものを用意したぞ。



CD-ROM収録先 A Magnavi Ip0001 Htmltips
今月号のTIPSをすべてCD-ROMに収録!!

このコーナーを楽しむために

最新のHTMLを使う際に、どうしても避けて通れないのがWWWブラウザの互換性の問題だ。そこでこのコーナーでは、TIPSごとにブラウザの対応状況をアイコンで表している(11月10日現在)。これを参考に使用するWWWブラウザを選んでほしい。



インターネットエクスプローラ3以上



インターネットエクスプローラ4以上



インターネットエクスプローラ5以上



ネットスケープナビゲーター3以上



ネットスケープナビゲーター4以上



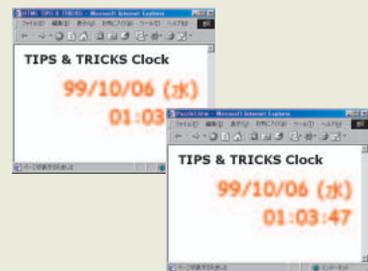
12月号「HTMLパズルに挑戦しよう」の解答

さまざまな解答が寄せられたが、数字が正確に表示されれば正解とした。スクリプトが長いので、解答のすべては掲載できない。CD-ROMに収録したソースを見てほしい。ここでは省略しているが、時計の文字列を表示させる<DIV>タグには「clock」というID名を付けてスタイルシートで位置を指定していることに注意。

ANSWER ① 時計を表示せよ!

Dateオブジェクトから時刻を取得し、IEならinnerHTMLで、ナビゲーターならレイヤーのdocumentオブジェクトを使って文字列を表示する。

```
function showclock(){
  days = new Array("日", "月", "火", "水", "木", "金", "土");
  d = new Date();
  year = d.getFullYear();
  month = d.getMonth() + 1; if (month < 10) month = "0" + month;
  (中略)
  str = year + "/" + month + "/" + date + "(" + days[d.getDay()] + ") " +
  hour + ":" + minute + ":" + second;
  if (document.all) clock.innerHTML = str;
  else if (document.layers) {
    document.clock.document.open();
    document.clock.document.write(str);
    document.clock.document.close();
  }
}
setInterval("showclock()", 1000);
```

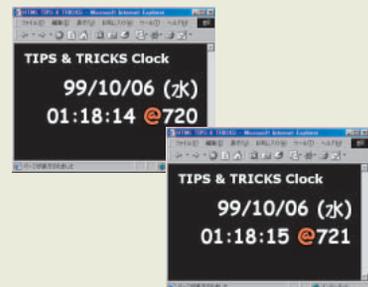


正解者: 村田正彦さん、谷口勝宣さん、よしともさん、稲見知志さん、齊藤貴志さん、うおまさ@homeさん

ANSWER ② インターネットタイムを表示せよ!

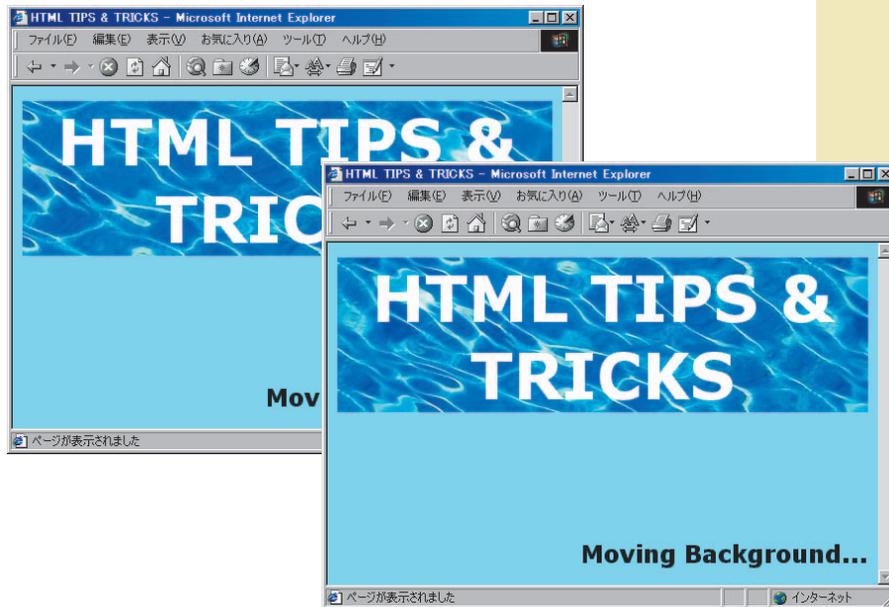
getUTCHoursなどのメソッドを使って世界標準時を取得し、beatを計算する。

```
beathour = d.getUTCHours() + 1; if (beathour > 23) beathour = 0;
beat = Math.floor((beathour * 3600 +
  d.getUTCMinutes() * 60 + d.getUTCSeconds()) / 86.4);
if (beat < 100) beat = "0" + beat; if (beat < 10) beat = "0" + beat;
str = year + "/" + month + "/" + date + "(" + days[d.getDay()] + ") " +
  hour + ":" + minute + ":" + second + "@" + beat;
```



正解者: 村田正彦さん、谷口勝宣さん、よしともさん、稲見知志さん、齊藤貴志さん、うおまさ@homeさん

背景画像を動かす



インターネットエクスプローラのダイナミックHTMLでは、スタイルシートのあらゆるプロパティをスクリプトで操作できる。フォントサイズやフォント名、文字色や背景色など変更できないものはない。ということは、背景画像もスタイルシートで指定すれば、動かせるようになるはずだ。左のサンプルを見てほしい。ページのタイトル文字の背景に敷かれた水面のような画像が流れるように動いている。アニメーションGIFを作らなくても、簡単なスクリプトだけでこれだけの効果を出せるのだ。難しいことをやっているように見えるが、この連載で今まで何度も出てきたテクニックを応用しているだけだ。



1

```
<STYLE TYPE="text/css">
H1 { text-align: center; color: white; font: bold 64px Verdana;
background-image: url(tips1.jpg); }
</STYLE>
```

2

```
<SCRIPT LANGUAGE="JavaScript">
var pos = 0;
function timer () {
title.style.backgroundColor = pos + " " + pos;
pos += 8;
}
</SCRIPT>
```

3

```
<BODY onLoad="setInterval('timer()', 500);">
<H1 ID="title">HTML TIPS & TRICKS</H1>
```

Point

サンプルのソースは、ひと目見ればわかるような簡単なものだ。まずソース①のスタイルシートと③の<H1>タグで、背景に画像を敷いた見出しを作る。もちろん<P>タグや<DIV>タグでも同じように背景画像を敷いて動かせる。<BODY>タグの背景画像、つまりページ全体の背景画像も動かせるので、試してみると面白いだろう。背景画像を動かすタグには、スクリプトで操作できるようにID属性(ここでは「title」)を付ける。

ソース③の<BODY>タグに設定したonLoadイベントでは、setIntervalメソッドを使ってアニメーション処理を行う関数「timer」が500ミリ秒(1/2秒)ごとに呼び出されるようにする。ソース②の関数「timer」を見てみよう。やっていることはごく簡単。「title」というID名のオブジェクトの背景画像を少しずつ動かしているだけだ。

背景画像の位置は、「style.backgroundColor」で取得したり変更したりできる。これはスタイルシートの「background-position」と対応していて、たとえば「30 20」という文字列を入れると、並べ始める位置が左から30ピクセル、上から20ピクセルに変更される。ほかに背景関連のプロパティには次のようなものがあり、いずれもスクリプトで変更できる。

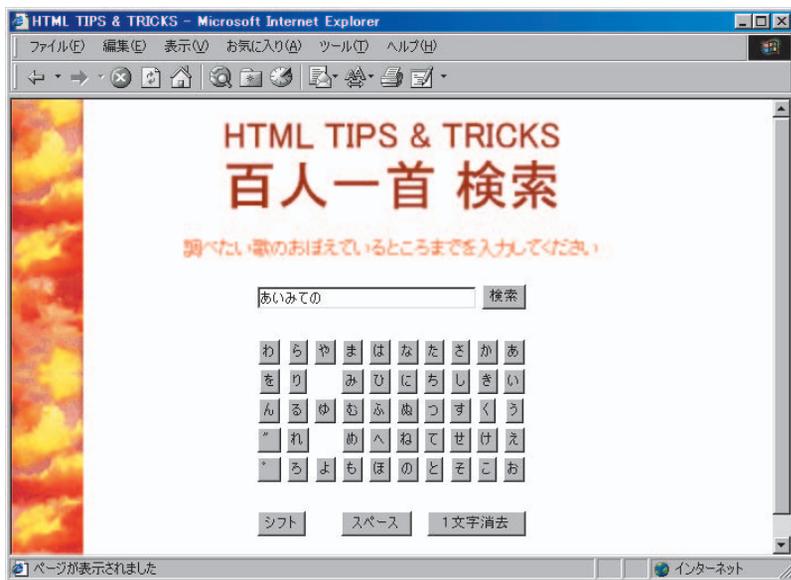
```
style.backgroundColor
背景画像を固定させるかスクロールさせるか
("scroll"または"fixed")
style.backgroundColor
背景色("#FF0000"など)
style.backgroundColor
背景の画像ファイル("url(ファイル名)")
```

```
style.backgroundColor
背景画像を並べる方向("repeat"、
"no-repeat"、"repeat-x"、"repeat-y")
```

このサンプルでは、「pos」という変数を用意して、「timer」が呼ばれるたびにその数を8ずつ増やしている。「style.backgroundColor」に「pos + " " + pos」という文字列を入れれば、500ミリ秒ごとに背景画像の開始位置が「8 8」,「16 16」,「24 24」,.....と動いていくことになる。

以上で背景画像を動かすスクリプトが出来上がる。実際に自分のページで応用するには、このサンプルのように継ぎ目がなめらかで、並べると連続しているように見える画像を使うと効果的だろう。

ソフトウェアキーボードを作る その2



1999年9月号でソフトウェアキーボードの作り方を紹介したが、もう使ってみたらどうか。斬新なアイデアで人目を引いたと思うが、かなり画面を広くとってしまっているのが気になった人もいるかもしれない。そこで今回は濁音と半濁音、それに「っ」や「ゃ」などの小さい文字の入力方法を工夫して、キーの数を26個減らす方法を紹介しよう。左のサンプルは、百人一首の中にある歌の最初の文字をひらがなで入力すると続きを検索してくれるフォームをイメージしたものだ。9月号に比べてコンパクトにできている。もう9月号を忘れてしまった人もいると思うが、この記事だけでもできるので心配は無用だ。

1

```
<SCRIPT LANGUAGE="JavaScript">
function enter (moji) {
  nstr = moji.value;
  str = moji.form.box.value;
  browser = navigator.appName;
  i = "あ".length;
  lstr = str.substr(0, str.length - i);
  rstr = str.substr(str.length - i, str.length);
  if (nstr == " ") {
    if (rstr == "か") { rstr = ""; nstr = "が"; }
    else if (rstr == "き") { rstr = ""; nstr = "ぎ"; }
    (中略)
    else { nstr = rstr; rstr = ""; }
  }
  (中略)
  if (nstr == "1文字消去") { rstr = ""; nstr = ""; }
  if (nstr == "スペース") { nstr = " "; }
  moji.form.box.value = lstr + rstr + nstr;
}
</SCRIPT>
```



2

```
<FORM>
<INPUT TYPE="text" NAME="box" SIZE="45">
<INPUT TYPE="submit" VALUE=" 検索 ">
<INPUT TYPE="button" VALUE="1文字消去" onClick="enter (this);">
<INPUT TYPE="button" VALUE="あ" onClick="enter (this);">
<INPUT TYPE="button" VALUE="い" onClick="enter (this);">
(中略)
<INPUT TYPE="button" VALUE="スペース" onClick="enter (this);">
<INPUT TYPE="button" VALUE=" " onClick="enter (this);">
<INPUT TYPE="button" VALUE="°" onClick="enter (this);">
<INPUT TYPE="button" VALUE="シフト" onClick="enter (this);">
</FORM>
```

POINT

まずソース②のフォームを見てみよう。各ボタンのVALUE属性に五十音の仮名が割り振られていて、ボタンを押すと、「enter」という関数にどのボタンを押したのかを知らせようになっている。ここまでで前回と同じようなキーボードができる。問題は濁音だ。

このサンプルでは、たとえば「が」を入力するときには、「か」ボタンの後に「 」ボタンを押すことにした。関数に「 」ボタンが押されたときに1つ前に入力された「か」を「が」に変える機能を追加しよう。ソース①の関数enterを上から見てみると、まず変数「nstr」に押されたボタンの文字(moji.value)を入れる。次にすでにテキストボックスにある文字列(moji.form.box.value)を変数「str」に代入しよう。続いてこの文字列の右端の文字(つまり1つ前に

入力した文字)を切り出すわけだが、ブラウザーによって動作が違うので、「i = "あ".length;」というおまじないの1行を入れておく。「lstr」は文字列の最初から2つ前に入力した文字までを入れ、「rstr」には右端の文字が入る。

nstrが濁点「 」で、かつrstrが濁点を付ける文字のときは、nstrに濁点付きの文字を入れてrstrを空(「 」)にする。if文を参照してほしい。か行、さ行、た行、は行の全20文字がその対象になる。もしもnstrが「 」なのにrstrが濁点を付けられない文字なら、nstrにそのままrstrを代入し、rstrを空にしておけばいい。

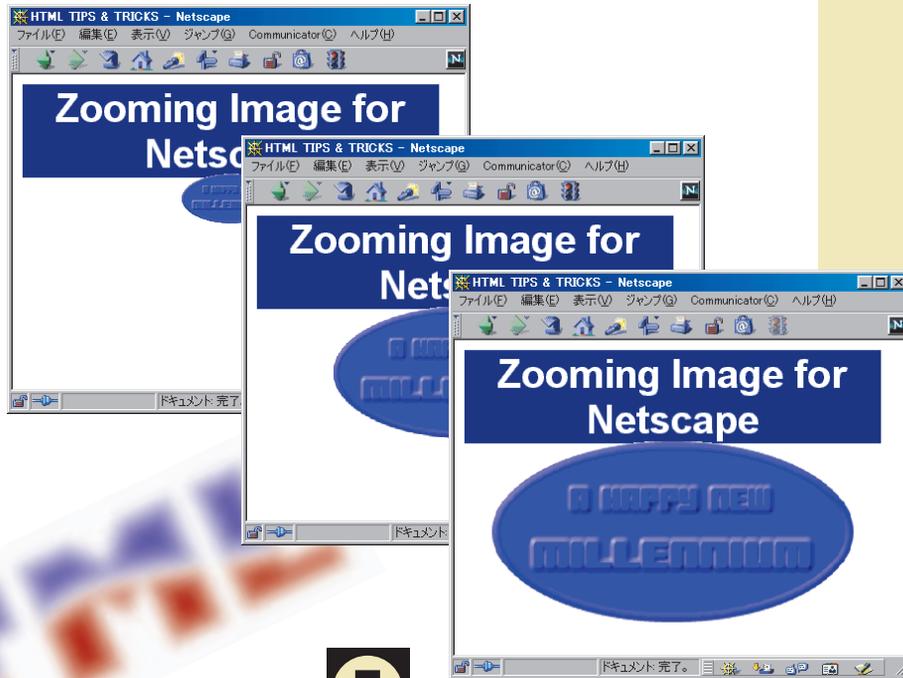
上記のソースでは省略しているが、半濁点「°」の場合も同じだ。は行の文字の場合にnstrに半濁点付きの文字を入れ、rstrを空にする。「シフト」

の場合は、あ行とや行、それに「つ」の小さい文字をnstrに入れ、rstrを空にする。1文字消去機能もあると便利だ。1つ前の文字を消すわけだから、rstrとnstrの両方を空にしておく。

最後にテキストボックスの文字を、lstr、rstr、nstrをつなぎ合わせたものにすれば出来上がりだ。もちろん、あらかじめ濁点や半濁点が付いた文字ボタンを作ってスクリプトを簡単に済ませることもできるが、画面の節約と遊び心を考えたら、こっちのほうがいいだろう。あとはボタンを配置すれば、ソフトウェアキーボード・バージョン2ができる。

なお、このサンプルは、実際に百人一首を検索する機能はないことに注意。

画像をズームアップさせる



1

```
var cnt = 1;
function zoom() {
  if ( cnt > 200 ) { clearTimeout ( timer ); }
  else {
    zmlmg = "<IMG SRC=zoom_img1.gif
      WIDTH=" + cnt * 2 + " HEIGHT=" + cnt + ">";
    document.zmField.document.open ();
    document.zmField.document.write (zmlmg);
    document.zmField.document.close ();
    cnt = cnt + 5;
    timer = setTimeout ("zoom()", 50)
  }
}
```

2

```
<BODY onLoad="zoom ()">
<DIV ID="zmField" STYLE="position:absolute; top:100px">
</DIV>
```

Point

このコーナーの読者なら、見慣れた部分が多くあるソースなので、それほど考え込むことなくスムーズに理解できるだろう。

ソース①から説明しよう。最初に変数「cnt」の初期値を1に設定している。この「cnt」は画像の幅と高さの基準となる値だ。関数「zoom」の中で、画像の幅と高さの指定に「cnt」を使っている。

次は関数zoomだ。ここではif～elseを使って画像の拡大と停止の処理を分けている。ifの条件は「cnt」の値で、このサンプルでは200を指定している。この値を超えるとスクリプトはclearTimeoutを実行し、画像の拡大が停止する。

次のelse文では、画像の大きさの設定とレイヤーの出力を行っている。画像の大きさは先にも触れたとおりcntの値を元に設定し、ここで得られた

タグを変数「zmlmg」に入れている。よく見るとWIDTHの値を「cnt * 2」としているが、これは今回のサンプルで使っている画像の縦横の比率が1対2になっているためだ。もし自分でオリジナルの画像を使う場合は、WIDTHとHEIGHTのどちらか一方の値を「cnt」にして、他方を「cnt * 比率」とするといいだろう。

次はレイヤーの出力の部分だ。ナビゲーターで新しいレイヤーを出力するには、これまでこのコーナーで何度も説明してきたようにdocumentオブジェクトの「open」「write」「close」を使う。この一連の流れは新しいレイヤーを出力する際には、必ず使うので覚えておこう。「zmField」とは、ソース②で示しているID名「zmField」の<DIV>タグで指定した領域のことなので、ここに画像が表示されることになる。次にcntの値に5を加えているが、これは関数zoomが次に呼び出されたとき

のcntの値になる。関数zoomが実行されるたびに画像が5だけ大きくなる。

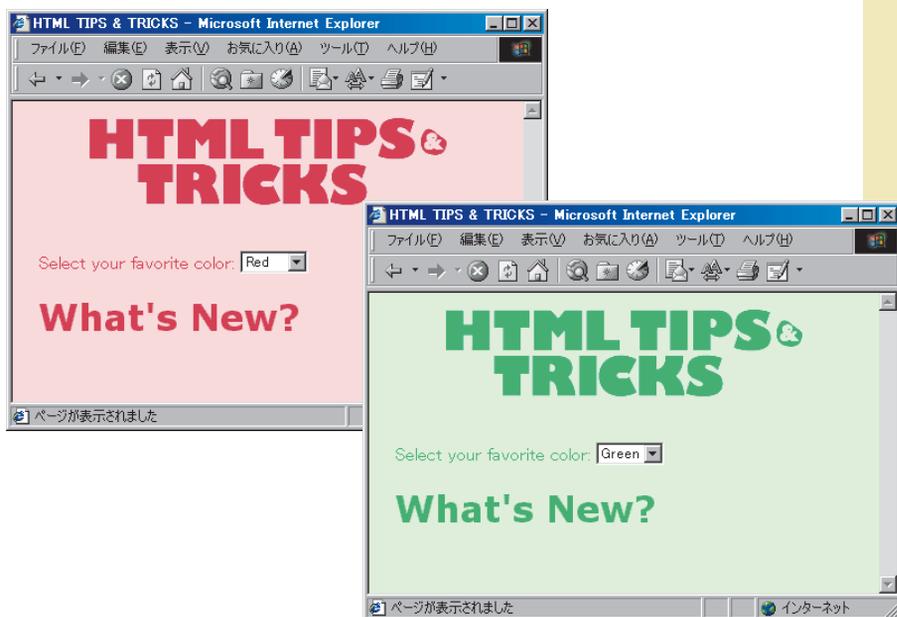
関数zoomの最後にsetTimeoutを使ってタイマーを設定する。このサンプルでは50ミリ秒(1000分の50秒)ごとに関数zoomを呼び出すようにしている。このタイマー設定があるので関数zoomが指定した時間ごとに実行され、画像がズームアップする演出ができるのだ。

ソース②を見てみよう。ここでは単純にID名を「zmField」と指定した空のレイヤーを作っているだけだ。この位置にスクリプトで作られた画像が出力される。必ずスタイルシートで位置指定(position:absolute)をしよう。これを設定しなければ画像は表示されないの注意してほしい。

N
4

1999年2月号で紹介した画像をズームアップさせるTIPSを覚えている読者は多いだろう。あのテクニックはエクスプローラのバージョン4以上でしか動作しないものだった。今回はナビゲーター4で動作するズームアップのTIPSを紹介しよう。左はそのサンプルだが、中央に表示している画像が徐々に拡大している様子わかるだろう。これはレイヤーを連続して出力させて、その中表示する画像の幅と高さをレイヤーと同じように連続して大きくすることで可能な演出なのだ。少し難しく聞こえるかもしれないが、ソースとポイントを比較しながらよく読んで実践すれば、意外と簡単だとわかるだろう。

カスタマイズできるページを作る



```
<SCRIPT LANGUAGE=" JavaScript">
var logo = new Array ("red.gif", "blue.gif", "green.gif");
var bcol = new Array ("#FFE0E0", "#D0E0FF", "#E0FFE0");
var tcol = new Array ("#CE3952", "#5273CE", "#42B57B");
tag = "<IMG SRC=";
if (location.search == "?red") {
    tag += logo [0];
    document.bgColor = bcol [0];
    document.fgColor = tcol [0];
}
else if (location.search == "?blue") {
    tag += logo [1];
    document.bgColor = bcol [1];
    document.fgColor = tcol [1];
}
}
```

```
else if (location.search == "?green") {
    tag += logo [2];
    document.bgColor = bcol [2];
    document.fgColor = tcol [2];
}
else {
    tag += logo [0];
    document.bgColor = bcol [0];
    document.fgColor = tcol [0];
}
tag += " WIDTH=320 HEIGHT=77">";
document.write (tag);
</SCRIPT>
```

POINT

こういう仕掛けを作ろうとするときにすぐ思いつるのは、クッキーを使って設定を保存し、次回にその設定を読み出してページを変更することだ。しかし、このサンプルではクッキーは使っていない。また、配色ごとにいちいちページを作っているわけでもない。HTMLファイルは1つだけだ。そのタネは実に簡単で、URLの中に色の指定を保存しているのだ。URLの最後に「?」を付けて、「index.html?red」のときは赤色の画面、「index.html?blue」では青色の画面、「index.html?green」では緑色の画面が表示されるようにしている。ファイル名に「?」を付けてオプションを追加するというのは、CGIプログラムにキーワードを渡す場合と同じだ。CGIだけでなくHTMLファイルを呼ぶときにもこの「ファイル名?何々」は使える。

HTMLファイルの中で「?何々」を取り出すには、

JavaScriptで「location.search」を調べればいい。これは昔からJavaScriptに備わっているので、IE3やナビゲーター3でも使える。上記のソースを見ればわかるように、この値が「?red」、「?blue」、「?green」の場合にそれぞれに対応する背景色と文字色を「document.bgColor」と「document.fgColor」に設定しているだけだ。また、それぞれの場合で画像ファイルの名前を変えて「IMG」タグを変数「tag」に組み立てて、「document.write」でTIPS & TRICKSのロゴを表示させている。

あとは、ユーザーがオプションを選べるような仕掛けを作ればいい。ここではプルダウンメニューを使ってURLを変えているが、CGIを呼び出すようなフォームを作成してACTION属性にHTMLファイルを指定することもできる。

上記のソースでは省略したが、ユーザーが色を選択するためのプルダウンメニューを見てみよう。これは、よく使われているスクリプトで、<SELECT>タグと<OPTION>タグでジャンプするものだ。

```
<FORM>
<SELECT onChange="location.href=
this.options [selectedIndex].value">
<OPTION VALUE="index.html?red">Red
<OPTION VALUE="index.html?blue">Blue
<OPTION VALUE="index.html?green">Green
</SELECT>
</FORM>
```

なお、このサンプルはローカルのハードディスク上では動作しないことに注意。FTPでサーバー上にアップロードしてからテストするようにしよう。



「My Yahoo!」のようなポータルサイトのサービスには、モジュールの位置やページの配色などをユーザーが指定できるカスタマイズ機能が備わっている。これをまねして、自分のページを訪れた人が好みの色を選べる仕組みを作ってみよう。左のサンプルは、初めて訪れたときは赤色を中心とした配色になっているが、ページ中央のプルダウンメニューから「Red」、「Blue」、「Green」のいずれかを選べば、ページの内容は同じままで画面の配色が切り替わる。切り替わったところでこのページをお気に入りやブックマークに入れると、次回からは選んだ配色で表示される。さっそくこの仕掛けを見てみよう。

HTMLパズルに挑戦しよう

隠されたトリックを解き明かせ！



大好評発売中！



今月のテーマ

・ミレニアムイベントを制する・

ついに人類は2000年という大きな節目を迎えることになる。このタイミングを狙って、自分のホームページで何かイベントを企画している読者は多いのではないだろうか。そこで今回は、そのイベントにふさわしいパズルを2問用意してみた。ヒントは先月号のクイズに隠されているので、挑戦しやすいはずだ。このパズルを解いて2000年という新しい時代へ訪問者を招待しよう。トリックがわかったらすぐに解答を送ってきてほしい。正解者には抽選で1名にオリジナル折りたたみ傘をプレゼントさせていただく。なお、正解は来月のこのコーナーで発表する。それでは頭をやわらかくして、今月のテーマ“ミレニアムイベントを制する”にチャレンジ！

「HTMLパズルに挑戦しよう」宛先

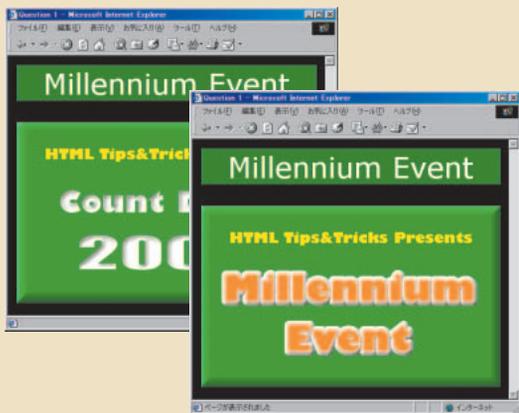
正解がわかった人も、わからなかった人も、ご意見、ご感想など何でもOK、次の宛先にメールしよう。用件の欄には必ずHTML TIPS & TRICKSの1行を忘れずに。あなたの挑戦を待つ！

✉ im-html@impress.co.jp

なお、締め切りは12月10日とさせていただきます。

QUESTION 1

2000年に画像を切り替える！



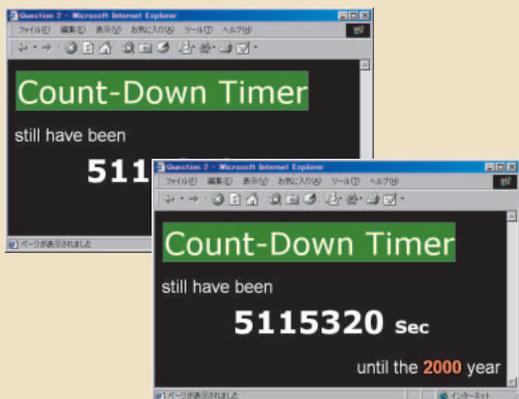
左のサンプルには2つの異なる画像が表示されている。これは2000年1月1日午前0時ちょうど前後で、訪れたときに表示される画像が異なるページを表したものだ。0時ジャストの前か後かを判断しなければならないのだから、時刻だけでも分だけでも秒だけでもいけない。JavaScriptには、ある時間を基準としてミリ秒を返す日付オブジェクトがある。解答を得るためには、このオブジェクトを使って2000年1月1日午前0時と現在時刻を比較すればいい。なお、このクイズはIE3でも動作するようにしなければならない。最近のTIPSに慣れている人は、それがネックになるかもしれないが基本に戻ろう。そうすれば解答が得られるはずだ。



現在時刻を1970年1月1日からカウントする日付オブジェクトは.....

QUESTION 2

2000年までカウントダウン！



左のサンプルは、1秒ごとに2000年1月1日午前0時までの時間をカウントダウンしている様子を表したものだ。先月号の第1問では現在時刻をテキストとして表示させる問題と載せたが、今月号の第2問の解答を表示させる仕組みも同じになっている。あとはどうやってカウントダウンさせるかなのだが、これは1997年11月号を参考にするといいだろう。「そんな古いバックナンバーなんて持ってないよ」という人は、「ホームページ裏技大全 HTML TIPS & TRICKS」の63ページを参照してもらおう。なお、このクイズの解答はEとナビゲーターのどちらか一方で動作するものであれば正解とする。



小数点以下を無視するには、Math.ceilやMath.floorがあるぞ.....



[インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

株式会社インプレスR&D

All-in-One INTERNET magazine 編集部

im-info@impress.co.jp