

HTML TIPS & TRICKS

第34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

先月発売されたこの連載の単行本「ホームページ裏ワザ大全 HTML TIPS & TRICKS」はもう購入したかな？ おかげさまで売れ行きは好調だ。単行本が発売されてもこの連載はまだまだ続くので、本を読みながら連載の新しいテクニックを毎月チェックすれば、自分の技をさらにパワーアップできるぞ。それでは今月もさまざまなTIPSとTRICKSをお届けしよう。



CD-ROM収録先 A Magnavi Ip9912 Hhtmltips
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(10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



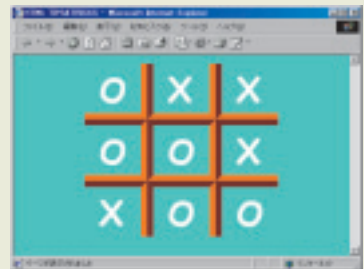
11月号「HTMLパズルに挑戦しよう」の解答

テーブルを使った基本的な技だったが、応募者が少なかったのは残念だ。第1問の2つの属性は、現在マイクロソフトのサイトのリファレンスにも掲載されていないためだろうか。HTML TIPS & TRICKSの単行本では、27ページの「好みの色で表の陰影を付ける」で紹介されている。

ANSWER ① 外枠のない表を作れ!

外枠のない表を作るには、<TABLE>タグのFRAME属性に「void」を指定する。枠線の明るい部分の色は「BORDERCOLORLIGHT」属性で、暗い部分の色は「BORDERCOLORDARK」属性で指定できる。さらにIE 4以降ではスタイルシートで「border-width」を指定すると枠線が太くなる。

```
<STYLE TYPE="text/css">
TD { color: white; border-width: 8px; padding: 2 24 24; }
</STYLE>
:
<TABLE BORDER FRAME="void"
BORDERCOLORDARK="#800000" BORDERCOLORLIGHT="#FF8000"
CELLSPACING=0 ALIGN="center">
```

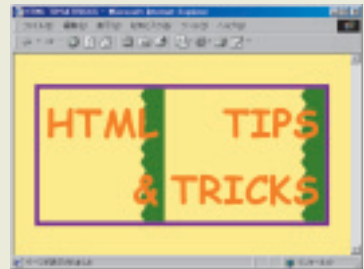


正解者：加藤秀一さん、よしともさん、うおまさ@homeさん、谷口勝宣さん、齋藤貴志さん

ANSWER ② セルの右側に画像を並べろ!

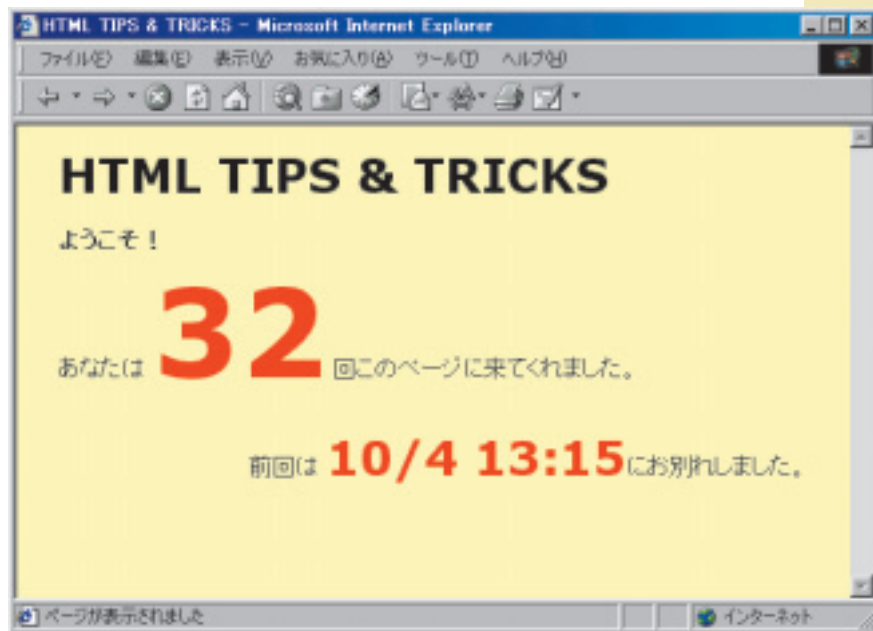
スタイルシートを使って<TD>タグにbackgroundプロパティで背景画像、並べる方向、並べる位置を指定する。画像ファイルは「url(ファイル名)」とし、縦方向にだけ並べるには「repeat-y」を、右側に並べるには「right」を指定する。

```
<STYLE TYPE="text/css">
TD { color: #FF8000; background: url(puzzle2.gif) repeat-y right; }
</STYLE>
```



正解者：加藤秀一さん、Kumeuchi Akiraさん、よしともさん、うおまさ@homeさん、谷口勝宣さん、齋藤貴志さん

訪問した回数を記録する



訪れた人の数を記録するアクセスカウンターを自分のページに付けている人は多いだろうが、訪れた人がそれぞれ何回ずつページを見たかを表示するには、クッキーを使わなければならない。しかし、CGIやJavaScriptでクッキーを扱うのはなかなか面倒だ。クッキーの構文をいちいち解析する必要があるからだ。ところが左のサンプルでは、クッキーを一切使わずにページを見た人の訪問回数を表示している。いったいどうやっているのだろう。IE 5で追加された新機能を使えば、訪問回数のようにページを表示したときのデータを記録して、再び表示したときにそのデータを取り出すことが簡単にできるのだ。



1

```
<STYLE TYPE="text/css">
SPAN.ud { behavior: url(#default#userData); }
</STYLE>
```

2

```
<BODY onLoad="loadCounter()"
onUnload="saveCounter()">
あなたは
<SPAN CLASS="ud" ID="oCounter">1</SPAN>
回このページに来てくれました。
```

3

```
<SCRIPT LANGUAGE="JavaScript">
function loadCounter () {
oCounter.load ("MyData");
v = oCounter.getAttribute ("count");
if (v) oCounter.innerText = Number (v) + 1;
}
function saveCounter () {
oCounter.setAttribute ("count", oCounter.innerText);
oCounter.save ("MyData");
}
</SCRIPT>
```

Point

ページ中のデータを保存するには、先月号の「指定のページをホームページにする」と同じくDHTMLビヘイビアを使えばいい。まずソース①のようにスタイルシートを使って、「SPAN.ud」つまり「ud」というクラス名のタグに機能を加える。behaviorプロパティに「url(#default#userData)」という値を指定すれば、データ保存機能が追加されることになる。

ソース②では、訪問回数を表示するためのタグにクラス名「ud」を指定してデータ保存機能を追加している。また、スクリプトで扱えるように「oCounter」というID名を付けた。ここでは表示用のタグとデータ保存機能付きのタグを同じにしているが、別のタグでもかまわない。<BODY>タグにはonLoadイベントとonUnloadイベントを指定して、ページを読み込んだときとページを閉じたときにスクリプトが呼び出されるようにする。

ソース③は、ページを読み込んだときに訪問回数を表示する関数「loadCounter」と、閉じたときに訪問回数を保存する関数「saveCounter」だ。まずsaveCounterから見てみよう。「setAttribute」はDHTMLビヘイビアによってタグに加えられたメソッドで、データを指定するものだ。引数にはデータの名前とデータの値を入れる。ここではデータ名を「count」とし、その値を「oCounter.innerText」とし、つまり現在表示している訪問回数とした。データの名前と値は次のようにいくつでも指定できる。

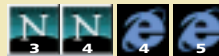
```
oCounter.setAttribute ("email",
"im-html@impress.co.jp");
oCounter.setAttribute ("nickname",
"TipsTricker");
```

次の「save」メソッドで指定したデータが実際に保存される。saveメソッドの引数「MyData」は、ひとまとまりのデータ全体の名前だ。

いったんこのページを閉じてから再びページを読み込むと、関数loadCounterが呼び出される。loadメソッドでデータ全体を読み込んだら、getAttributeメソッドで「count」のように名前を付けたデータを1つずつ読み出せる。取り出したデータに1を加えてタグのinnerTextに入れれば、訪問回数が表示される。データの値は文字列なので、数値に変換してから計算すること。

上記のソースには載せていないが、前回ページを閉じたときの日付も記録して表示させている。CD-ROMに収録したソースを見てみよう。

マウスの動きで3つの画像を入れ替える



TIPS & TRICKSの読者にはおなじみのテクニックの1つに、マウスカーソルの移動で画像を入れ替えるというものがある。最近ではリンク用の画像や個人のデジカメギャラリーなどでも定番のテクニックになってきた。そこで今回は、このテクニックにもうひと工夫加えて、3つの画像を順番に入れ替える方法を紹介してみよう。左のサンプルでは、写真の上にマウスカーソルを載せたりカーソルをはずしたりするたびに、3枚の写真が順に切り替わって表示される。2枚の画像を入れ替える例ならよく見かけるが、3枚ならぐっと新鮮な効果が出せる。マンネリになってきた君のギャラリーもこれでデザイン一新だ。



1

```
<SCRIPT LANGUAGE="JavaScript1.1">
imag = new Array ("tips2a.jpg", "tips2b.jpg", "tips2c.jpg");
im = new Array ();
for (var i = imag.length; i > -1; i--) {
  im [i] = new Image ();
  im [i].src = imag [i];
}
var j = 1;
function rounding (imgname) {
  imgname.src = im [j % 3].src;
  j = j + 1;
}
</SCRIPT>
```

2

```
<A HREF="index.htm"
  onMouseOver="roundimg (photo);"
  onMouseOut="roundimg (photo);">
<IMG SRC="tips2a.jpg" NAME="photo"></A>
```

POINT

画像を入れ替えるテクニック自体は、画像が2枚の場合とまったく同じだ。まず、画像がきれいに表示されるように、あらかじめ3枚の画像をブラウザに読み込ませておこう。ソース①を見てほしい。使いたい3個の画像ファイルの名前を配列「imag」に覚えさせたら、それを使って画像の配列「im」を作り、すべての画像を読み込ませる。

「im [i] = new Image ();」で画像オブジェクトを準備し、「im [i].src = imag [i];」で画像のファイル名を当てはめればよい。この方法は画像2枚を切り替えるときにも使える技だ。こうしておくと、切り替えたときにダウンロードが始まって×印アイコンが表示されることがなくなる。

続いて画像3枚を扱う場合の特有のソースを説明しよう。変数「j」は、ちょっとしたおまじないだ。初期値を1にしておこう。次に「roundimg」とい

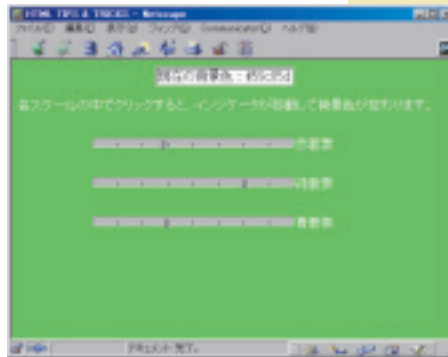
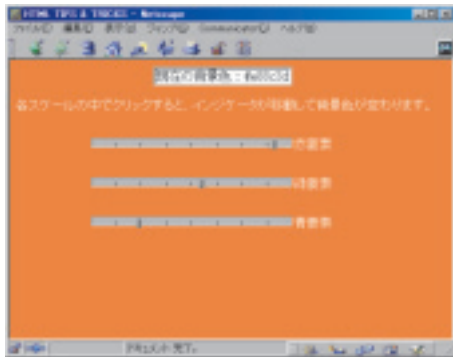
う関数を作る。引数の「imgname」は、関数を呼び出すときに対象となるタグの名前を入れることになる。関数の中の1行目で「imgname.src」に先ほど作った配列imの1つを当てはめる。ここでのポイントは、「im [j % 3]」でjの値によって違う画像が当てはめられることだ。「%」は「～で割った余り」という意味の演算子だ。

関数の2行目を見てほしい。jに1を加えている。変数jは、ページが開かれてからずっと生き残っているものなので、関数roundimgが呼ばれるたびに、jの値は1ずつ大きくなっていくことがわかるだろう。jを3で割った数値は、「1、2、0、1、2、0、……」と順番に変化する。つまり「imgname.src」には、配列imの3つの画像が、順番に当てはめられることになるわけだ。

最後に、入れ替えの対象にするソース②のタグを見てみよう。SRC属性には、配列imに最初に入れた画像のファイル名を書いておきたいだろう。名前は適当に(ここでは「photo」)付ける。サンプルでは「onMouseOver」と「onMouseOut」の2つのイベントで関数roundimgを呼び出している。これでマウスが画像の上に乗ったりはずれたりしたときに、画像が順番に入れ替わる。

この方法のよいところは、一度onMouseOverで表示された画像は、次回はonMouseOutで表示されることだ(なぜなら3は奇数だから)。2枚を入れ替える場合のように、右クリックでの画像の保存ができなくなることがない。もちろんonMouseOverのときだけ画像を入れ替えるようにするのもいいだろう。

背景色を自在に設定する



1

```
<FORM NAME="colField">
<INPUT TYPE="text" NAME="col" SIZE="20">
</FORM>
<LAYER ID="redScale" TOP="100" WIDTH="255">
<IMG SRC="scale.gif" WIDTH="255" BORDER="0">
</LAYER>
<LAYER ID="redIndicator" TOP="100" onLoad="init()">
<IMG SRC="indicator.gif" BORDER="0">
</LAYER>
```

2

```
function init() {
  document.layers["redScale"].captureEvents
    (Event.MOUSEBUTTONDOWN | Event.MOUSEUP);
  document.layers["redScale"].onmousedown = redMove;
  document.layers["redScale"].onmouseup = setbgCol;
}

function redMove(event) {
  var r = event.layerX;
  document.layers["redIndicator"].moveTo(r,100);
}
```

```
function toHex(n) {
  if (n < 0) return "00";
  else if (n > 255) return "ff";
  else return Math.floor(n / 16).toString(16) +
    (n % 16).toString(16);
}

function setbgCol() {
  var r = document.layers["redIndicator"].left;
  document.bgColor = "#" + toHex(r) + "0000";
  document.colVal.col.value =
    "現在の背景色: " + document.bgColor;
}
```

POINT

まず最初にお断りしておく。上記の画面では赤、緑、青の各色の要素のインジケータを表示しているが、ソースに関しては誌面の都合により赤色の処理だけ掲載した。説明も赤色の部分だけとするが、ほかの色の処理も基本的に同じやり方なので、このコーナーの読者ならすぐに理解できるだろう。

まずはソース①だ。関数「init」ではマウスイベントの設定をしている。今回はマウスボタンを押したとき(onmousedown)と上げたとき(onmouseup)のそれぞれのイベントで別の関数を呼び出すように指定している。

関数「redMove」はマウスボタンを押したときに呼び出される関数で、レイヤーで作ったインジケータを移動している。

関数「toHex」は関数「setbgCol」から呼び出されるので、これについてはあとで説明する。関数「setbgCol」を先に説明しよう。ここでは、ID名を「redIndicator」と付けたレイヤーの左座標を変数「r」として扱っている。redIndicatorレイヤーは「redScale」レイヤーの上を移動するので、rの値は0から255までの間で変化することになる。次の文は背景色を設定する処理で、関数toHexに変数rを渡したときの戻り値を設定している。最後に背景色を文字列として確認できるように、1行テキストボックスに書き出している。

それでは関数toHexの中身を見てみよう。この関数では、関数setbgColから渡された引数を元に条件処理をして、結果を16進数にして返している。注目してほしいのは、最終的に「document.bgColor」を設定するのに使うので、計算結果を文

字列にしていることだ。ある数値を文字列にした場合は、数値に「toString()」を付けなければならないが、ここでは16進数なので0~fが得られるように「toString(16)」としている。

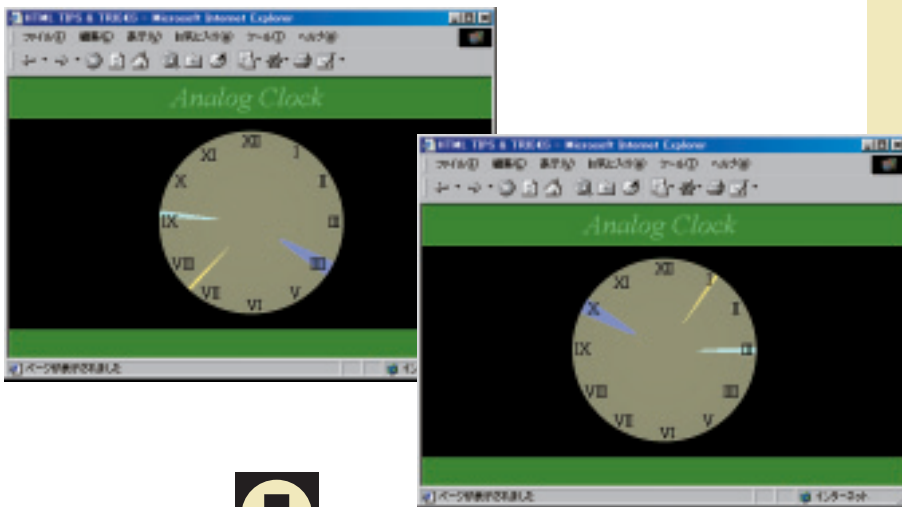
ソース②は、スケール用の画像とインジケータ用の画像を載せる2つのレイヤーを指定している。ID名「redScale」のレイヤーと、載せている画像「scale.gif」の幅を255に指定していることがわかるだろう。これはインジケータの移動領域になるので、必ず255を指定しよう。

以上が赤色を処理するソースの説明だ。緑色と青色を含めたこのスクリプトの全ソースは、本誌の付録CD-ROMに収録しているので参照してほしい。

N
4

みなさんはページの色を決めるときに何を参考にするだろうか。ウェブページや参考書などの色見本を参照して色を決定していると思う。しかし、そうした色見本はせいぜい216色程度しか掲載していないものがほとんどだ。ご存じのとおり、HTMLではRGBの指定によって $16(0\sim f)$ の6乗(約1677万)の数の色が使える。とは言っても、テキストエディターで色を指定するときは、少し薄い赤色を使おうと思って「fa0000」と書いてもその色をイメージできるだろうか？ 結局は実際にブラウザ上で色の確認をすることになる。そこで今回はページを製作する際にぜひ使ってほしいTIPSを紹介しよう。

アナログ時計を作る



1

```
var r = 100;
var conv = (2 * Math.PI)/360;

function rotate () {
  var now = new Date();
  var degSec = now.getSeconds() * 6 - 90;
  var degMin = now.getMinutes() * 6 - 90;
  var degHour = now.getHours() * 30 - 90;

  var xSec = r + r * Math.cos(degSec * conv);
  var ySec = r + r * Math.sin(degSec * conv);
  var xMin = r + r * Math.cos(degMin * conv);
  var yMin = r + r * Math.sin(degMin * conv);
  var xHour = r + r * Math.cos(degHour * conv);
  var yHour = r + r * Math.sin(degHour * conv);

  clock.filters[0].moveLight(1,xSec,ySec,0,1);
  clock.filters[0].moveLight(2,xMin,yMin,0,1);
  clock.filters[0].moveLight(3,xHour,yHour,0,1);

  timerID = setTimeout('rotate()', 1000);
}
```

2

```
function init () {
  clock.filters[0].addCone(111,108,1000,104,104,
    155,155,155,100,300);
  clock.filters[0].addCone(111,108,0,0,0,155,155,0,100,1);
  clock.filters[0].addCone(111,108,2,0,0,0,255,255,100,2);
  clock.filters[0].addCone(111,108,5,0,0,0,0,255,100,5);
  rotate();
}
```

3

```
<IMG SRC="clock.gif" ID="clock" onLoad="init()"
  STYLE="height: 200px; width: 200px; filter:light()">
```



このコーナーの読者なら、自分のホームページに現在の時刻を表示する時計を付けている人は少なくないだろう。それでは「その時計はどんなタイプ?」と聞かれば、「デジタル時計」と答えるだろう。そこで今回は、ホームページではまったく見かけることのない、アナログ時計を作るテクニックを紹介しよう。先月号のこのコーナーでレーザー画面のようなアニメーション表示をするスクリプトを紹介したが、あのテクニックはこのアナログ時計を説明するための導入でしかなかったのだ。ここでは先月号のスクリプトを大幅に改造して「アナログ時計」という、古典的でありながら斬新な演出に挑戦してみよう。

POINT

先月号のレーザーアニメーションとソースが似ていることに気付いただろう。先月号では光源を単純に回転させていただけだったが、今回は光源の回転を現在の時刻(時、分、秒)の値を基準として処理してみた。これがアナログ時計を作る発想の源だ。

まずはソース①から説明しよう。変数「r」と「conv」は先月号と同じで、それぞれ光源の中心の値と角度の数値を変換するのに使う値だ。

関数「rotate」では、最初に時、分、秒を得て、それぞれの値を角度に変換して変数「degSec」、「degMin」、「degHour」に入れている。さらに、この値を元にサインとコサインを計算し、光源が照射する対象になる座標「xSec」、「ySec」、「xMin」、「yMin」、「xHour」、「yHour」を決めている。

「moveLight」メソッドがある行では、ID名「clock」の画像の上で光源が照射する方向を移動させている。かつこ内に光源が照射する先の座標を入れれば、時計、分針、秒針を模した光が移動するようになる。かつこ内の引数の詳細については先月号を参照してほしい。

関数「rotate」の最後では、setTimeoutを使って1000ミリ秒(1秒)ごとに自分自身の関数(rotate)を呼び出すようにしている。関数「rotate」が呼び出されるたびに現在の時刻を取得するので、それに伴って角度の値が変わり、最終的に光を照射する方向も変わるのだ。

次はソース②を見てみよう。関数「init」はソース③の画像が読み込まれたとき一度だけ実行されるもので、「addCone」メソッドで順に「画像全体を照らす光源」、「秒針を模した光源」、「分針を模

した光源」、「時計を模した光源」をID名「clock」の画像に追加している。addConeメソッドの引数についても、先月号で説明しているので参照してほしい。

光源をすべて設定し終わったあとに関数rotateを呼び出しているのが、光源の照射する方向を移動させる関数に処理が移るのだ。

最後はソース③なのだが、これは説明するまでもなく普通のタグにスタイルシートとID名を設定して、画像を読み込んだとき(onLoad)に関数「init」を呼び出しているだけだ。

以上でアナログ時計スクリプトは完成だ。さっそく自分のホームページに組み込んで、訪問者を驚かせてみよう。

HTMLパズルに挑戦しよう

隠されたトリックを解き明かせ!



今月のテーマ

時計を制する

前ページではIEのフィルター機能を使ってアナログ時計を作成した。時計のスク립トは、ホームページを訪れる人に強い印象を与えることができるし、スク립トの腕だめしにもちょうどいい。前ページのような派手な時計でなくても、ページの片隅に簡単な時計を表示させておいてはどうだろうか。そこで今月は、刻一刻と現在の時刻を表示し続けるデジタル時計の作成に挑戦してみよう。トリックがわかったらすぐに解答を送ってほしい。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“時計を制する”にチャレンジ!

「HTMLパズルに挑戦しよう」

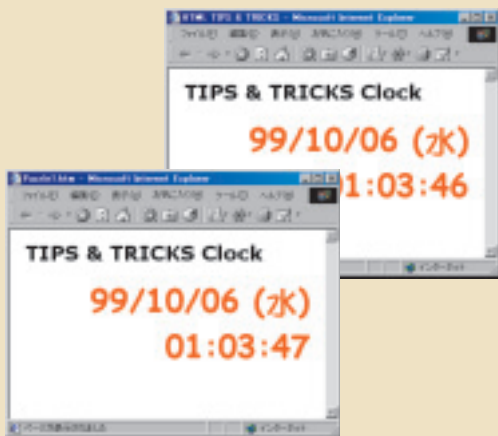
宛先

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11月10日とさせていただきます。

QUESTION 1 時計を表示せよ!

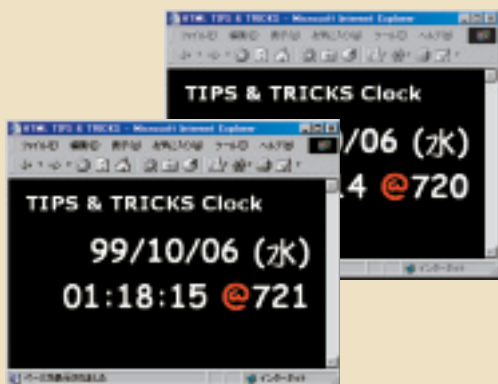


左のサンプルは、1秒おきに現在の時刻を表示する時計をページに埋め込んだものだ。テキストボックスを使った時計ならよく見かけるが、ここでは普通のテキストとして表示された時刻が変化している。こうしたデジタル時計を作るのが第1問だ。解答の条件は、月、日、時、分、秒を2桁で表示すること(1桁の時は0で始まること)、曜日を表示すること、テキストボックスやステータスバーなどは使わないこととする。IE 4以降とナビゲーター4のダイナミックHTMLを使えば簡単だが、数値の処理はちょっと面倒かもしれない。なお、第1問と第2問の解答は、IEとナビゲーターのどちらか片方でしか動かないものでかまわない。



6月号の2番目のTIPSのテクニックを使って……

QUESTION 2 インターネットタイムを表示せよ!



第1問は年月日と時分秒を表示するだけの単純な時計だったが、左のサンプルではそれに加えて@マークの後ろに3桁の数字が表示されている。これは最近流行の「インターネットタイム」を表したものだ。インターネットタイムとは、Swatch社が提案している単位「beat」を使った時刻の表現だ。1日の1000分の1を1 beatとし、Swatch社の所在地の午前0時(世界標準時からは+1時間で、日本との時差は-8時間)を世界共通の0 beatとする。たとえば日本では午前8時が0 beat、午後2時が250 beat、午後8時が500 beatとなる。このインターネットタイムを自分のページに表示させて、新し物好きな人を驚かせてみよう。



1 beatは86.4秒、JavaScriptには世界標準時を得るメソッドもある……



[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