

HTML TIPS & TRICKS

第 47 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 / 高橋登史朗

ついにネットスケープ6が正式版になった。多くの人が新しいバージョンに乗り換えるようになれば、サイト制作者はバージョン4のスタイルシートのバグにもう悩まずに済みそうだ。一方でネットスケープ6の元になっているMozillaは、終わることなく進化を続けている。それでは今月の最初のTIPSでネットスケープ6の新機能をさっそく味わってみよう。



CD-ROM収録先 ☐ Magnavi Ip0101 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4以上



インターネットエクスプローラ5以上



インターネットエクスプローラ5.5以上



ネットスケープナビゲーター4以上



ネットスケープ6以上



Mozilla(数字はMilestoneの番号)



12月号「HTMLパズルに挑戦しよう」の解答

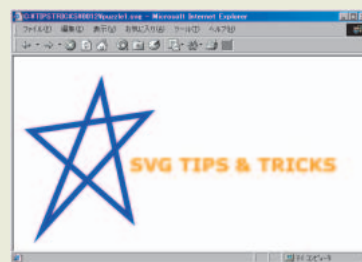
今月の挑戦者はやや少なめだった。英語のサイトを参照するのはきつかったらうか。SVGの仕様はとにかく膨大なので、テキストエディターだけで記述するのは面倒だが、W3Cやアドビのサイトをざっと眺めるだけでも興味深い機能がいくつも見つかるはずだ。



ANSWER 1 星を描け!

第1問では<polygon>タグを使い、points属性に頂点の座標をスペースで区切って順番に並べていくと星が描ける。SVGで線を描くには、<polygon>のほかにも、今月の2番目のTIPSで紹介する<path>タグを使うこともできる。

```
<svg>
<polygon style="fill:none;stroke:#4444CC;stroke-width:10"
points="150,50 50,300 230,100 30,130 200,280" />
</svg>
```



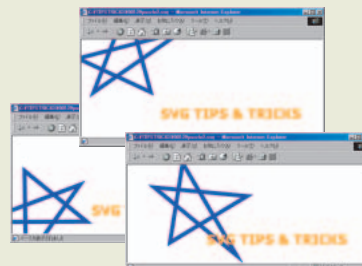
正解者: 坂部和久さん、ENDEさん、Masahiko Murataさん、富園慎一郎さん、山口雅仁さん、よしともさん、sionさん



ANSWER 2 星を回転させる!

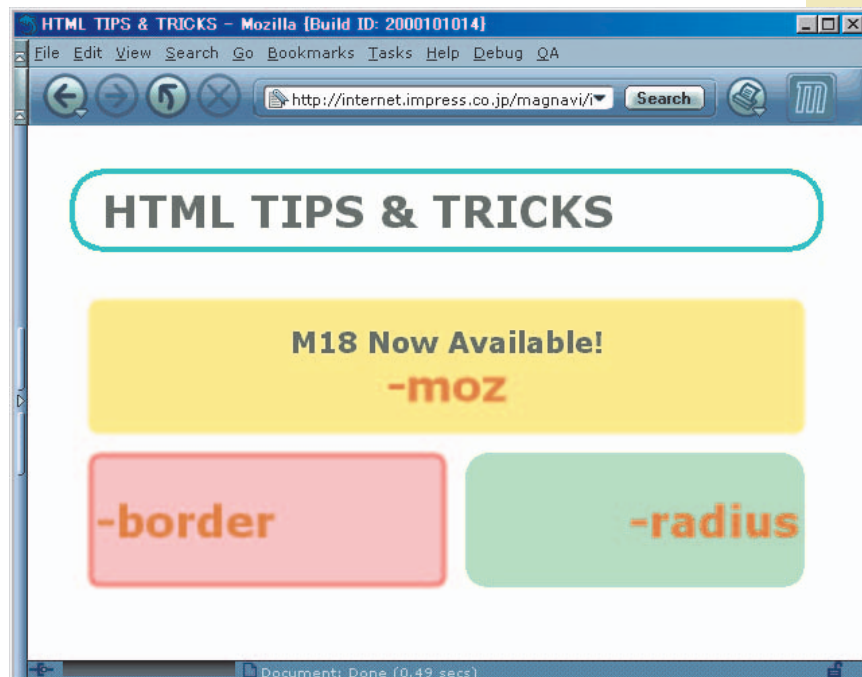
<polygon> ~ </polygon>の中に<animateTransform>タグを置く。星の真ん中が回転の中心になるように、<g transform="translate(.....)">で原点をずらす。

```
<svg>
<g transform="translate(150 150)">
<polygon style="fill:none;stroke:#4444CC;stroke-width:10"
points="0,-100 -50,150 80,-50 -120,-20 50,130">
<animateTransform attributeName="transform"
type="rotate" values="0;360" dur="2s" repeatDur="indefinite"/>
</polygon>
</g>
</svg>
```



正解者: 坂部和久さん、ENDEさん、Masahiko Murataさん、富園慎一郎さん、山口雅仁さん、sionさん、うおまさ@homeさん

角の丸い囲みを作る



テーブルを使って囲みを作っているページはよく見かける。なかには、4つの角に画像を使って囲みを丸く見せているページもある。しかし、そんなテクニックを使うと意味のない画像を乱用した重たいページになってしまいがちだ。意味のない画像にはアクセシビリティ上の問題も多い。左のサンプルを見てほしい。角を丸く見せた囲みがたくさんある。それなのに、このページでは画像を一切使っていない。Mozillaとネットスケープ6で用意された新しい技なのだ。ページデザインに新しい可能性をもたらすこのテクニックをさっそく紹介しよう。（編集部）



```
<STYLE TYPE="text/css">
.kakomi {
border-style: solid; border-width: 4px;
border-color: #FFFF99; background-color: #FFFF99;
-moz-border-radius: 10; }
</STYLE>
```

Point

この技は、先月号の「文字や画像を半透明にする」と同じく、Mozillaやネットスケープ6のスタイルシート拡張機能を使ったものだ。先月は「-moz-opacity」というプロパティで透明度を指定したが、今回は「-moz-border-radius」を使う。このプロパティの値に数値を指定すると、角の丸みの半径になる。数値を大きくすればするほど、丸みの部分が大きくなる。とはいっても、まずは囲みの線が引かれるようにしておかないと角の丸みを指定しても意味がないので、これまで何度も使ったことのあるプロパティで枠線を指定しておく。「border-style」で線の種類を、「border-width」で線の太さを、「border-color」で線の色を指定するのだ。

```
border-style: solid; border-width: 4px;
border-color: #FFFF99;
```

この3つは、「border」プロパティで1つにまとめられることは、ご存知のとおり。

```
border: solid 4px #FFFF99;
```

枠線だけを引いてもいいが、「background-color」を使って枠線と背景色を同じにしておけば、違った雰囲気のある囲みが作れる。

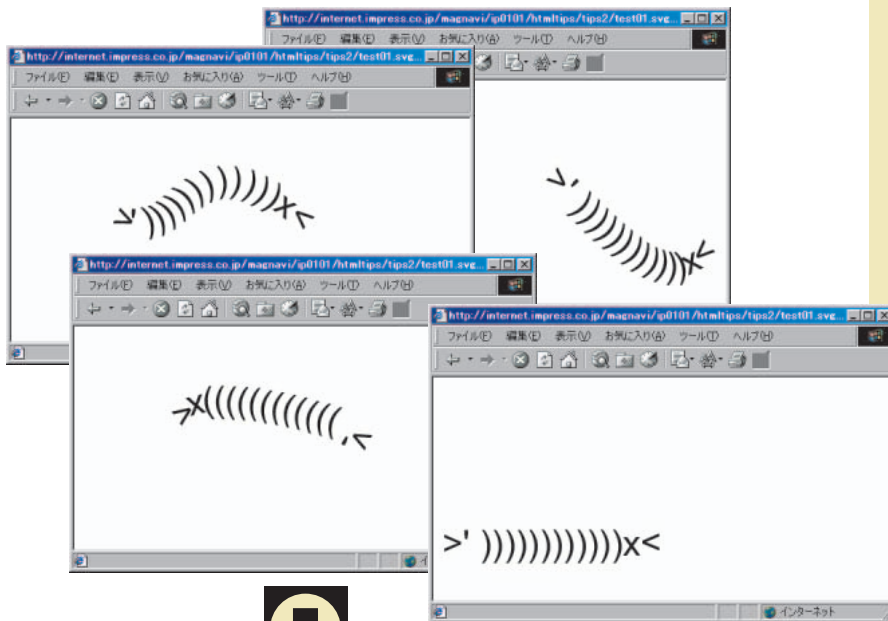
「-moz」で始まるプロパティは、Mozillaやネットスケープ6全体のインターフェイスをスタイルシートで設計できるように作られたものだ。サンプルのように<H1>や<TD>などのタグに指定できるだけでなく、フォームの部品に指定して、ボタンやテキストボックスの角も丸くできる。たとえば、<STYLE>タグに次のようなスタイルを追加すると、送信ボタンの角がすべて丸くなる。

```
INPUT [type="submit"] {
-moz-border-radius: 5; }
```

この「~」という指定は、CSS 2で導入された「属性セレクター」という新機能で、「type="submit"」という属性を持つ<INPUT>タグに限定してスタイルを変更できる。こんなところからも、Mozillaやネットスケープ6がEよりもはるかにCSS 2対応が進んでいることがわかるだろう。

なお、ここでMozillaやネットスケープ6の新バージョンに関する注意。1999年9月号から11月号のXULのサンプルにあった「openDialog」メソッドは、セキュリティ強化のために使えなくなった。XUL ファイルを開くときのスクリプトは、「window.open("01.xul", 'x1', 'width= 200, height=200)」のように書き換えてほしい。

SVGで魚をくねくね泳がせる



```
<?xml version="1.0" ?>
<svg>
<path id="p" style="fill:none"
  d="M 10 200
    C 500 200 500 200 300 100
    C 10 10 100 200 200 100
    C 300 10 400 300 500 200" />
<text id="test" style="font-family:Arial; font-size:40px; fill:#000000">
<textPath xlink:href="#p">
  >' ) ) ) ) ) ) ) ) ) ) x<
<animate attributeName="startOffset" attributeType="XML"
  begin="0s" dur="3s" fill="freeze" from="100%" to="0%" />
</textPath>
</text>
</svg>
```



前はSVGで文字をぐるぐる回したが、今回は移動させてみよう。顔文字の要領でテキストで描いた魚を好きな曲線に沿って自由に泳がせてみる。こうした動きはSVGではいろいろな方法で書けるのだが、今回もなるべくシンプルな方法で書いてみることにした。ソースを見れば分かるが、`<xml>`と`<svg>`タグを含めてもわずかに6個のタグしか使っていない。スクリプトを使わずに、これだけでページ全体をくねくねと動き回ってお魚さんが描けてしまうのだから驚異的だ。もちろん、サンプルと同じ絵文字である必要はないから、自分の好きな文字に変えて動かしてみよう。（高橋登史朗）

POINT

このアニメーションの仕組みは、`<text>`タグの中に文字が書かれた「>')))))))))) x<」の絵を、`<path>`タグで指定した曲線に沿って、`<animate>`タグで動かすというものだ。

最初の`<path>`タグの中の「fill: none」は、塗りつぶしのスタイルだ。「none」は塗らないという意味になるので、指定したカーブは見えなくなっている。試しに「red」などの色を指定してみよう。曲線の内側が塗りつぶされて現れるはずだ。「d」属性はパスのカーブを指定している。

```
d="M 10 200 ....."
```

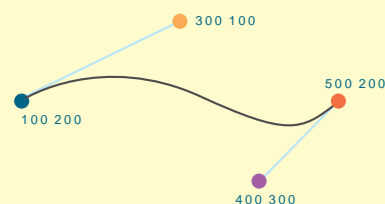
d属性は「パスデータ」と言って、`<path>`タグが描く図形のデータを定義する。英文字のMやCはコマンドで、M(moveto)を使って「M 10 200」と書

くことで、パスが書き出す図形の基点のx座標を10、y座標を200にするという意味になる。C (curveto)は「C 300 10 400 300 500 200」のように使って、ベジェ曲線(x1 y1 x2 y2 x y)を定義する。Mで指定した点や前のCの終点が基点になり、x yを終点、x1 y1とx2 y2を制御点としたカーブが描かれる。ベジェ曲線はグラフィックソフトの曲線を描くツールを想像すると分かりやすい。基点の間に置かれたいくつかの点(制御点)を元に滑らかな曲線を求める方法だ。

`<textPath>`タグでは、属性に「xlink:href="#p」と書くことで、「p」というIDの`<path>`タグで定義するカーブに`<text>`タグをリンクさせている。

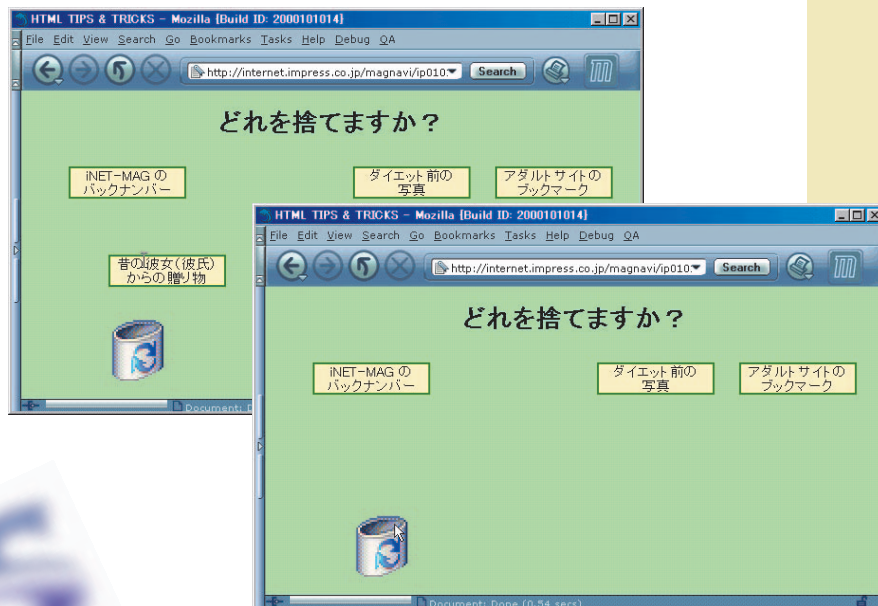
`<animate>`タグでは魚の絵文字を動かすわけだが、「from="100%" to="0%"」は`<path>`タグで指定した最後の点からMで指定した最初の基点まで動かすという意味だ。

```
d="M100 200 C300 100 400 300 500 200"
で描いたベジェ曲線
```



`<path>`タグのd属性のコマンドには、ほかにもL(直線) H(水平線) V(垂直線) A(円弧)などいろいろあるので、仕様を確認しながら動きをいろいろに変えてみると楽しいぞ。

ページ上にごみ箱を作る



ウィンドウズやマッキントッシュのユーザーなら、ファイルの移動などの作業で「ドラッグアンドドロップ」操作はお馴染みだ。ここで紹介するTIPSは、この操作をブラウザー上で実現するスクリプトだ。左のサンプルは、デスクトップでよく使われる「ファイルをごみ箱に捨てる」作業をブラウザーの画面上でまねしたもので、枠付きの文字をごみ箱の上にドラッグアンドドロップすると、その文字が画面上から消えるようにしてある。スクリプトは多少長めになっているので、尻込みしてしまうかもしれないが、ポイント欄の説明をよく読んでチャレンジしてほしい。

(大内勇)

```
<SCRIPT TYPE="text/javascript">
var g_obj;
function beginDrag (e, id) {
  x = e.clientX, y = e.clientY;
  g_obj = document.getElementById (id);
  window.onmousemove = doDrag;
  window.onmouseup = endDrag;
}
function doDrag (e) {
  var dX = e.clientX - x, dY = e.clientY - y;
  var newX = parseInt (g_obj.style.left) + dX + "px";
  var newY = parseInt (g_obj.style.top) + dY + "px";
  g_obj.style.left = newX;
  g_obj.style.top = newY;
  x = e.clientX, y = e.clientY;
}
```

```
function endDrag (e) {
  window.onmousemove = null;
  if (((e.clientX > 50) && (e.clientX < 200)) &&
    ((e.clientY > 240) && (e.clientY < 340)))
    g_obj.style.display = "none";
}
</SCRIPT>
<DIV STYLE="position: absolute;
top: 300px; left: 100px">
<IMG SRC="trashbox.gif"></DIV>
<DIV ID="obj1"
onmousedown="beginDrag (event, 'obj1');"
STYLE="position: absolute; left: 70px; top:
80px; width: 120px">
iNET-MAG の<BR>バックナンバー</DIV>
```

POINT

順番が逆になるが、先にソースDから説明しよう。最初の<DIV>タグはごみ箱の指定で、次の<DIV>タグは枠付き文字の指定だ。それぞれ、スタイルシートによって位置を指定している。また枠付き文字のほうはマウスを押したときのイベント「onmousedown」を設定して、関数「beginDrag」へマウスの位置情報 (event) とID名の2つの引数を渡すようにしている。

次にA～Cのスクリプトの説明だ。Aの関数「beginDrag」では、Dのマウスを押したときのイベントで渡された2つの値を使い、変数「x」(マウスポインタの横座標) と「y」(同縦座標) として「g_obj」(対象オブジェクト) を設定している。変数を設定したあとに、マウスを動かしているとき(ドラッグ中)は関数「doDrag」を呼び出し、マウスのボタンから指が離れたとき

(ドロップ)は関数「endDrag」を呼び出すようにしている。なお「対象オブジェクト」とは、ID名を「obj1」や「obj2」と付けた<DIV>タグで囲った要素全体のことだ。

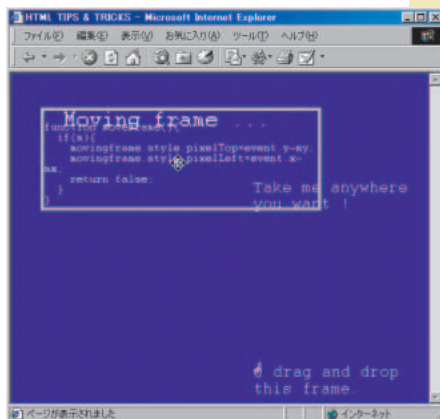
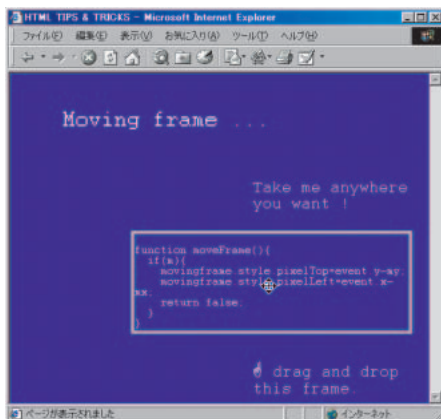
Bの関数「doDrag」はドラッグ中の動作を行う。変数「dX」と「dY」はマウスの移動量を表し、移動量に移動前の対象オブジェクトの左上座標の値を加算したものが変数「newX」と「newY」になる。ここで「parseInt」に注目してほしい。「g_obj.style.left」と「g_obj.style.top」の値は数字で得られるが、これは「100px」のような文字列であって数値ではない。ここでは加算や減算をするので文字列だと都合が悪い。そのため「parseInt」を使って文字列を整数として扱っているのだ。この変数「newX」と「newY」の値を対象オブジェクトの左上座標の値にするので、マウスボタンを押している間は、対象オブジェ

クトがマウスに追従するようになるのだ。

Cの関数「endDrag」ではドロップしたときの動作を行う。「window.onmousemove=null」とは、マウスが動いていても何もさせない処理。次のif文ではマウスの位置によって条件を分岐させている。その条件とは、ごみ箱の領域の座標で、「マウスポインタの横方向の座標が50から200ピクセルで、縦方向の座標が240から340ピクセルの場合」だ。この条件が成立しているときにif文の中が実行され、対象オブジェクトはディスプレイ上から消えてしまう。消えてしまうと言っても、スタイルシートで対象オブジェクトを表示させないようにしているだけだ。

以上で今回のスクリプトは完成だ。CGIと組み合わせれば、実際にサーバーにあるファイルを消去もできるので、興味のある人は試してみよう。

フレームを自在に動かす



1

```
<BODY bgcolor=indigo onLoad="pageLoad()">
<DIV ID="movingframe" STYLE="position:absolute;
height:128px; width: 348px; top: 256px; left: 256px;
border: 4px solid #CCCCC;
cursor: move; z-index:2;"
onmousedown="checkMouseDown ()">.....
```

2

```
<SCRIPT Language="JavaScript1.2">
var m = 0, mx = 0, my = 0;
function pageLoad () {
document.onmousemove = moveFrame;
document.onmouseup = checkMouseup;
}
function moveFrame () {
if (m) {
movingframe.style.pixelTop = event.y - my;
movingframe.style.pixelLeft = event.x - mx;
return false;
}
}
}
```

```
function checkMouseDown () {
if (event.srcElement == movingframe) {
m=1;
my = event.y-movingframe.style.pixelTop;
mx = event.x-movingframe.style.pixelLeft;
} else { return true; }
}
function checkMouseup () {
m=0;
}
}</SCRIPT>
```

タイトル、目次、ニュース、リンク、それに広告。ウェブを作り始めてしばらく経つと、トップページに次々と増える情報をどうやって整理して表示するか、それが問題になる。これまでもプルダウンメニューやポップアップメッセージなど、狭い画面を広く使うTIPSをいくつか紹介してきたが、新たに「自在に動かせるフレーム」を作ってみよう。ブラウザーの中に別のウィンドウがあるかのように位置を自由に変えられるフレームだ。たとえばフレームに最新ニュースを表示して、画面全体にある普段のコンテンツを読みたいときには、フレームをドラッグする、といった使い方ができる。 (藤井幸孝)

POINT

フレームといっても<FRAME>タグは使っていない。①のように<DIV>タグにスタイル「border: 4px solid #777777」を当てはめ、フレームのように見せかけている。またこの<DIV>には、「ID="movingframe"」と名前が付けてあり、スクリプトで操作できるようになっている。注意するのは位置指定のスタイルだ。「position: absolute」と絶対指定にしておこう。「z-index: 2」は、重なり合った複数の<DIV>の中で、この<DIV>が2番目にあることを表している。ついでだが、通常のウィンドウに似せて、ドラッグ中はマウスカーソルが十字マークになるように、「cursor: move」としておくと面白いだろう。

スクリプト部分は、マウスボタンの上下とドラッグ動作という「イベント」を感じて動くように作っている。ページを読み込んだときに呼ばれるソー

ス②の関数「pageLoad」では、このページ中でマウスが動かされたときにはいつも関数「moveFrame」が起動し、マウスボタンが離されたときには「checkMouseup」関数が動作するように仕掛けがしてある。checkMouseup関数では、変数「m」を0にセットしているが、このmはマウスボタンの状態を記憶するためのもので、0ならボタンがクリックされていない状態を、1ならクリックされてボタンが押されている状態を表している。

①の<DIV>タグの中に「onmousedown="checkMouseDown ()"」と書いておき、右マウスボタンを押すと、関数「checkMouseDown」が起動することに注目しよう。checkMouseDownでは、マウスボタンが押された場所がmovingframeオブジェクト(<DIV>タグ)かどうかを確認し、そうであればmを1にセットして、カーソルがフレームの中のどのあたりにあるかを算出している。「event.y -

movingframe.style.pixelTop」は、フレームの上端とカーソル位置の垂直方向の距離だ。水平方向も計算しておく。

このあとでカーソルが動かされると、最初の仕掛けによって関数moveFrameが起動し、mが1のとき、つまりボタンが押されればなしの間、フレームの位置を、現在マウスがある位置に移動させる。関数checkMouseDownで計算しておいた「my」や「mx」を使っているのは、フレームの中でのカーソルの位置を一定に保って、ドラッグの動きを自然に見せるのに役立っている。

適当なところでマウスボタンを離すと、関数checkMouseupによってmが0にセットされるので、関数moveFrameはフレームを動かさなくなり、ドラッグアンドドロップが完了するわけだ。どうだろう。普段のウィンドウ操作とまったく同じに扱えるはずだ。

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

・インラインフレームを制する・

ページの中に別のページを埋め込むインラインフレームは、IEの独自機能だったが、HTML 4.0に採用され、Mozillaとネットスケープ6でもサポートされた。今まであまり使う機会がなかったインラインフレームだが、2大ブラウザの標準機能となったことで、いろいろ工夫してみると便利な使い方が見つかるだろう。特にページ中の部品の再利用に役立ちそうだ。そこで今月はインラインフレームを活用したパズルに挑戦していただく。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“インラインフレームを制する”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

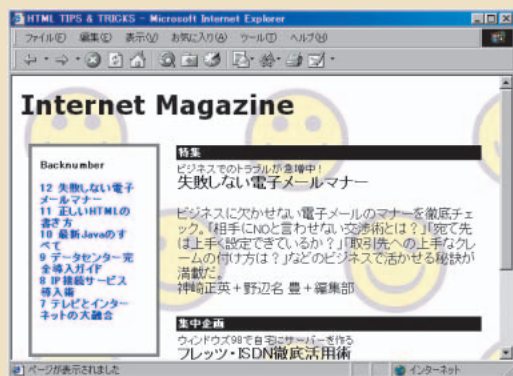
の1行を忘れずに。あなたの挑戦を待つ！

✉ 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 メニューを埋め込み！



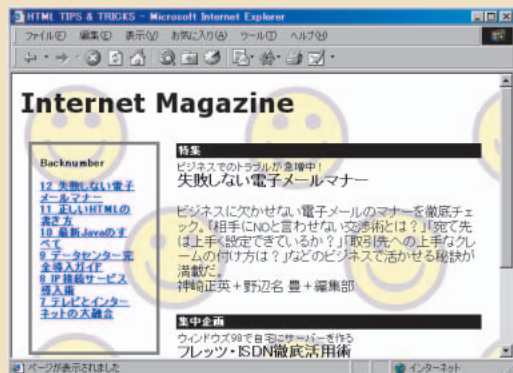
まずは基本的な使い方から。インラインフレームで何と言っても便利なのは、一度書いたHTMLを複数のページで何度も使い回せることだ。たとえば、共通するメニュー部分を1つのHTMLファイルに書いて、インラインフレームで埋め込めば、メニューを変更するときにはファイル1個を変更するだけで済む。しかも普通のフレームと違ってページごとのタイトルやURLがはっきりするので、訪問者にとって使いやすいページになる。そこで左のサンプルのようなリンクメニューを作ってみよう。単純にインラインフレームを指定するだけでなく、サンプルのようにサイズを固定させて、平べったい枠線を付けるのが問題だ。



お馴染みのスタイルシートを使う……



QUESTION 2 背景を連続させよ！



第2問は、第1問で作ったインラインフレームのメニューにIE 5.5の独自機能を追加して、面白い効果を出してみようというものだ。左のサンプルを見てほしい。枠に囲まれた部分はインラインフレームなのだが、なぜかウィンドウ全体の背景画像が透けて見えている。親となるHTMLとフレーム内のHTMLの両方に同じ画像を指定しているわけではない。親となるHTMLの背景画像がフレームの中にも連続しているのだ。さていったいどうすればいいのだろうか？ 2つのHTMLにちょっとした仕掛けをすればいいだけだ。あれこれ悩まずに、マイクロソフトのWeb Workshopに直接あたってみよう。



msdn.microsoft.com/workshop/author/dhtml/overview/inlinefloatingframes.asp ……

[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的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