

HTML TIPS & TRICKS

第48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

今月よりネットスケープ6の日本語正式版を付録CD-ROMに収録。今まで使ってみたことのなかった人も、これを機会に2000年6月号からのTIPS & TRICKSのバックナンバーを読んで、最新のスタイルシートやDOMの機能を試してみたいかがだろうか。それでは今月も、まだ誰も見たことのない最新のテクニックの数々を紹介しよう。



CD-ROM収録先 Magnavi Ip0102 Htmlltips
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(12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)



1月号「HTMLパズルに挑戦しよう」の解答

それほど難しい問題ではなかったが、挑戦者が少なかったのは残念だ。2問目でIE 5.5の新機能を英語のサイトで調べるのが面倒だっただろうか。今月のパズルでも、簡単な問題と難しい問題を1問ずつ用意したので、簡単なものだけでもチャレンジしてほしい。

ANSWER 1 メニューを埋め込め!

<IFRAME>タグにスタイルシートで枠線の種類、太さ、色の指定をするだけ。なお、インターネットエクスプローラでは、「FRAMEBORDER="0"」を指定しないと、フレームに立体的な枠が付いてしまう。

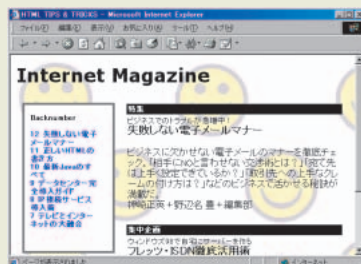
```
<IFRAME NAME="menu" SRC="menu.html"
FRAMEBORDER="0"
STYLE="border: solid 4px gray; width: 150px; height: 250px;">
</IFRAME>
```

ANSWER 2 背景を連続させよ!

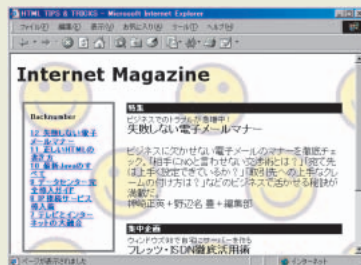
フレームを埋め込むほうのHTMLでは、<IFRAME>タグに「ALLOWTRANSPARENCY」属性を指定する。埋め込まれるほうのフレームのHTMLでは、<BODY>タグのスタイルシートで、背景色を「transparent」にする。

```
1 <IFRAME NAME="menu" SRC="menu.html"
ALLOWTRANSPARENCY="true" FRAMEBORDER="0"
STYLE="border: solid 4px gray; width: 150px; height: 250px;">
</IFRAME>
```

```
2 <BODY STYLE="background-color: transparent">
```

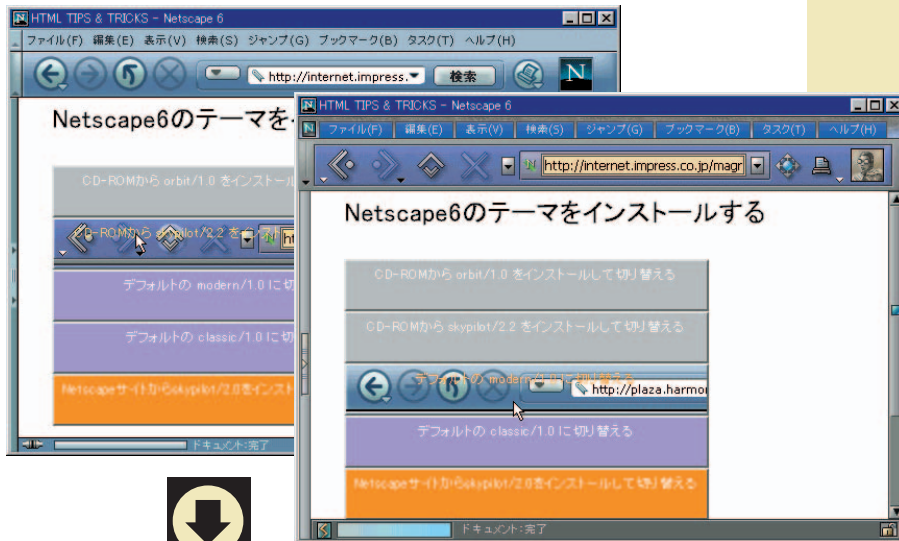


正解者: 坂部和久さん、Masahiko Murataさん、富園慎一郎さん、ENDEさん、山口雅仁さん、よしともさん、堀江さん



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

テーマをインストールする



1

```
<SCRIPT TYPE="text/javascript">
function chgSkin (swt, file, name) {
  var skinpath = new Array ();
  skinpath[0] = 'ftp://ftp.netscape.com/pub/netscape6/themes/english/6.0/';
  skinpath[1] = 'resource:/chrome/';
  skinpath[2] = 'chrome/';
  if ( navigator.userAgent.indexOf ("Netscape6") != -1 ) {
    InstallTrigger.installChrome (InstallTrigger.SKIN, skinpath[swt] + file, name);
  }
}
</SCRIPT>
```

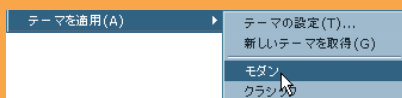
2

```
<INPUT TYPE="button"
VALUE="Netscapeからorbit/1.0をダウンロードして切り替える"
onclick="chgSkin (0, 'orbit.jar', 'orbit/1.0')"><BR>
<INPUT TYPE="button"
VALUE="デフォルトのmodern/1.0に切り替える"
onclick="chgSkin (1, 'modern.jar', 'modern/1.0')">
```

ネットスケープ6やMozillaの面白さの1つは、自分でブラウザをカスタマイズできることだ。しかも、改造できることにはいろいろなレベルがあり、プログラムに触れない初心者からディープなマニアまで、だれでも楽しめる仕掛けが組み込まれている。今回は、まずどんな人でもできる簡単なカスタマイズだ。「テーマ」と呼ばれる外観のデザインを、好みに合わせてインストールするボタンを作ってみよう。Linuxのウィンドウマネージャーのように、好きなものを選んでインターフェイスを取り替えられる機能だ。自分で作ったオリジナルテーマの配布もできてしまうぞ。（高橋登史朗）

Point

スクリプトを見る前に、まずテーマの切り替えを試してみよう。ネットスケープ6にははじめからテーマが2つ入っている。「モダン」と「クラシック」だ。メニューから「表示」「テーマを適用」で確認できる。



このリストからテーマの名前を選択すると、ブラウザ全体のデザインが魔法のように切り替わってしまうのだ。

テーマの種類はこの2つだけではない。「表示」「テーマを適用」「新しいテーマを取得」でネットスケープのテーマパークサイトへ行けば、いろいろなデザインのテーマが登録されていて、好き

なテーマをダウンロードして使えるようになってくる。テーマパークにあるいくつかのテーマはコンテストで一般から公募したもので、誰でも作れるものなのだ。テーマのソースはこの連載でも取り上げているXULでできており、ボタンなどのGIF画像を取り替えるだけでもカスタマイズはできてしまう。やる気のある人はネットスケープのサイト「Jump」を見てぜひ挑戦してみしてほしい。

それではソースを見てみよう。ボタンを押して関数「chgSkin」を起動すると、ネットスケープ6の場合は指定されたテーマをダウンロードして切り替えるというものだ。ダウンロード元はchgSkinの最初の引数で指定している。0、1、2でそれぞれFTPのディレクトリー、ハードディスク内の決められたフォルダー、CD-ROMのTIPS & TRICKSのフォルダーからインストールするようにしている。

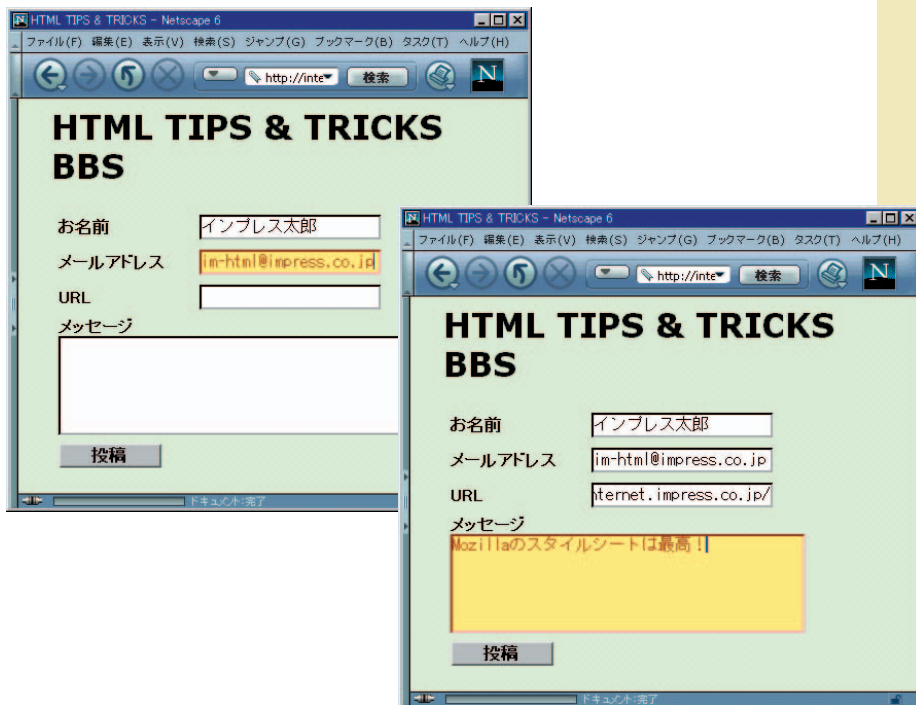
もし、自分のオリジナルテーマを作ったときは、配列「skinpath」の代わりに自分のテーマファイルへのパスを書き換えばいい。

if文の「navigator.userAgent.indexOf ("Netscape6") != -1」で、ブラウザがネットスケープ6の場合にだけ、続く{ }の中の命令が実行される。その命令が「InstallTrigger.installChrome」だ。InstallTriggerはインストールのためのオブジェクトで、installChromeメソッドにテーマのURLと名前を指定すると、OSに関係なく自動インストールを実行してくれる。

さあ、とりあえずはCD-ROMから好きなスキンをインストールしてさくさく切り替えて使ってみよう。

[Jump home.netscape.com/themes/createtheme.html](http://home.netscape.com/themes/createtheme.html)

入力欄に合わせて色を変える



1998年にCSS level 2がW3Cの勧告になったにもかかわらず、これまで2年以上の間、インターネットエクスプローラもネットスケープナビゲーターもCSSのサポートを十分にしていなかった。そこへ登場したのが、標準規格を完璧にサポートすることを目標としたMozillaとネットスケープ6だ。左のサンプルを見てみよう。掲示板のテキスト入力欄に書き込もうとするたびに、現在入力中の欄の背景色が変わる。今この欄に書いているのが分かりやすいし、使っていて楽しいページになる。CSS 2では、スクリプトを使わずに、ほんの数行の指定だけでこれができる。 (編集部)



```
INPUT[TYPE="text"]:focus, TEXTAREA:focus {
  background-color: #FFFF80; color: #AA0000;
  border-color: #FF0000;
}
```

POINT

テキストボックスやボタンなどのフォームの部品がキーボードで操作できる状態になっていることを「フォーカスが当たっている」という。CSS 2では、フォーカスが当たっている場合にスタイルシートを適用する仕組みがある。それには、これまで何度か紹介したことのある「擬似クラス」を使う。たとえばよく使われているものに、マウスをリンクの上に載せたときにフォントや色が変わるようにする「:hover」擬似クラスがある。

```
A:hover { font-weight: bold; }
```

これと同じようにCSS 2には、フォーカスが当たっている場合を表す「:focus」擬似クラスがある。次のような指定をすると、ボタンやテキストボックスにフォーカスが当たっているときの文字色を変えられる。

```
INPUT:focus { color: red; }
```

しかし、これだけの指定ではボタンとテキストボックスで別の色やフォントを使いたくても、同じになってしまう。そこで、先月の「角の丸い囲みを作る」のポイント欄で紹介した「属性セクター」を使って属性の値でスタイルを変えれば、ボタンとテキストボックスを区別できる。

```
INPUT[TYPE="text"]:focus {
  color: red; }
INPUT[TYPE="submit"]:focus {
  color: blue; font-weight: bold; }
```

CSSでは、(~)の前に指定する部分を「セクター」と呼ぶが、CSS 2ではセクターの機能が大幅に拡大されているので、一度仕様書をなが

めてみるといいだろう。たとえば次の例では、<H1>の次に<H2>が並んでいる場合にだけ、<H2>に対してスタイルが適用されて、見出しの間の余白を小さくできる。

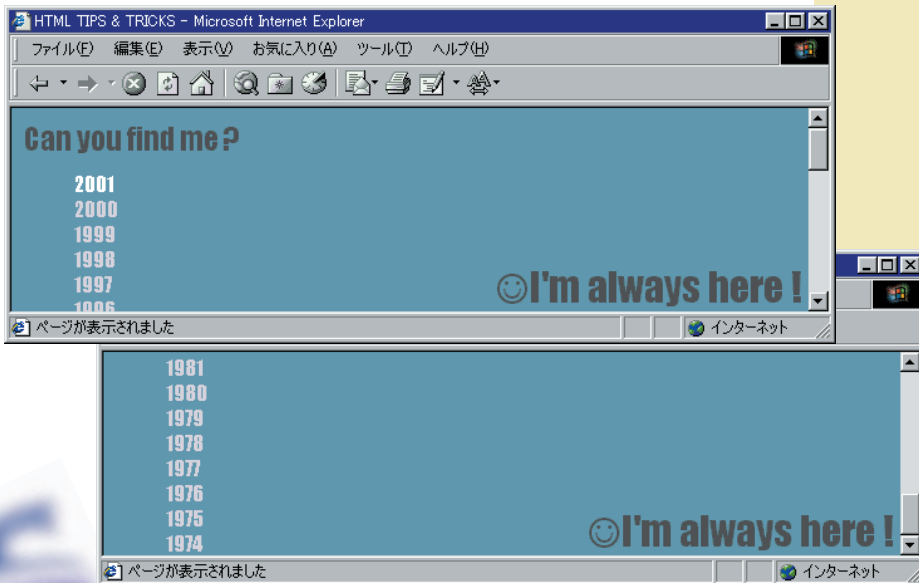
```
H1 + H2 { margin-top: -12pt; }
```

同じ名前の要素に自動的にテキストや画像を挿入する「:before」擬似要素と「:after」擬似要素は、2000年9月号で紹介した。

```
P:before { content: " "; }
```

CSS 2が普及すれば、ページ作成の手間を省いたり、新しいデザインの可能性を試したりできる。インターネットエクスプローラでも早く完全対応してほしいものだ。

文字や画像をつねに右下に表示する



バナー画像がいつもブラウザの右下に表示されているページを見たことがあるだろうか？ ウィンドウのサイズを変えたりスクロールさせたりすると、画像が移動してつねに右下の位置をキープするようなページだ。スタイリッシュなページを作っても、見る人の環境次第で崩れてしまうのはつらいと思っているデザイナーには、おすすめのTIPSだ。今回はシンプルに文字を固定してみた。画面をはじめに表示したとき、ウィンドウのサイズを変えたとき、スクロールバーを動かしたとき、「I'm always here!」という文字がいつもウィンドウの右下にあるように見える。

(藤井幸孝)

```
<SCRIPT LANGUAGE="JavaScript1.2">
var rb_width, rb_height;
function initialize () {
if (document.all) {
rb_width = rb.offsetWidth;
rb_height = rb.offsetHeight;
setInterval ("rb_ie()", 50);
rb.style.visibility = "visible";
}
else if (document.layers) {
rb_width = document.rb.document.width;
rb_height = document.rb.document.height;
setInterval ("rb_ns()", 50);
document.rb.visibility = "show";
}
}

```

```
function rb_ie () {
rb.style.pixelLeft = document.body.scrollLeft +
document.body.clientWidth - rb_width - 5;
rb.style.pixelTop = document.body.scrollTop +
document.body.clientHeight - rb_height;
}
function rb_ns () {
document.rb.left = window.pageXOffset +
window.innerWidth - rb_width - 30;
document.rb.top = window.pageYOffset +
window.innerHeight - rb_height;
}
window.onload = initialize;
</SCRIPT>

```

POINT

このスクリプトでは、次の数値を計算して、右下端のちょうどいい位置を割り出している。

スクロールの位置 + ウィンドウのサイズ
- <DIV>(オブジェクト)のサイズ

「スクロールの位置」について少し説明しておこう。分かりやすいのは縦方向の例だ。スクロールバーを動かしていくと、ページの上の方の部分はウィンドウの上にとんどん消えていく。スクロールの位置とは、ページの中のどこがウィンドウの一番上に来ているかを表すものだ。スクロールの位置に現在のウィンドウのサイズを足すと、ウィンドウの下端がページの中でどの位置にあたるかが分かるわけだ。この位置からオブジェクトの高さを引いて、その数値をオブジェクトの上下の座標として指定すれば、どんなときにもオブジェクトはウィン

ドウの下端に見えるという仕組みだ。上記のソースでは省略したが「I'm always here!」を囲った<DIV>タグに「rb」というID属性を付け、スタイルシートでpositionをabsoluteにして、自由に配置できるようにする。また、visibilityをhiddenにして隠しておき、スクリプトが起動するとvisibleに切り替わって表示されるようにする。

IEとナビゲーターで使用するプロパティなどがまったく異なるので、両方のブラウザ用に別々の関数を作っておく。関数「initialize」では、「document.all」があるならIE用の関数「rb_ie」を、「document.layers」があるならナビゲーター用の関数「rb_ns」をおなじみのsetIntervalで呼び出す。オブジェクトのサイズを割り出す方法も違っている。IEでは「オブジェクト名.offsetLeft(またはoffsetHeight)」を、ナビゲーターなら「docu

ment.オブジェクト名.document.width(またはheight)」を使う。

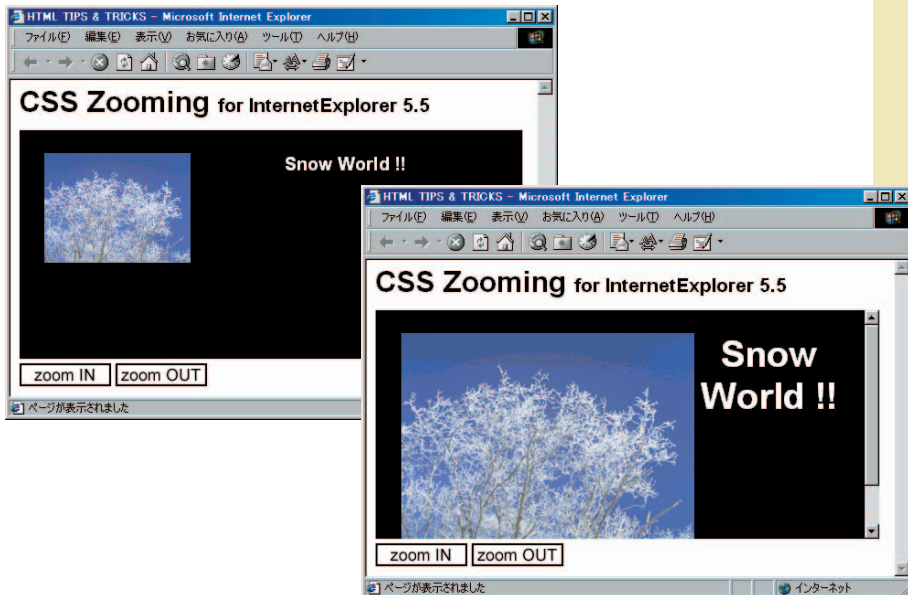
関数rb_ieと関数rb_nsでは、オブジェクトの位置を動かしている。2つの関数では、オブジェクトの左右の座標、上下の座標、スクロール位置、ウィンドウの内側の大きさを得る方法が異なるのでソースを確認しよう。横方向の位置を計算するとき5や30などの数字を引いているのは、そのままでは文字が右に寄り過ぎるケースがあるからだ。試してみて適当な数字を探すといいだろう。

最後に「window.onload = initialize」として、ページを開いたときに関数が呼ばれ、最初の描画がうまく動作するように仕掛けておく。テキストを1行だけ作ったつもりが2行に折れてしまうという場合は、なつかしの<NOBR>タグで回避できるので、うまくいかなかった人は試してみよう。

領域をまるごとズームさせる



5.5



以前、JavaScriptで画像の高さと幅を変化させてズーム効果を出すTIPSを紹介したが、一度に1つの要素にしかならなかつた。今回は、IE5.5から使えるようになったスタイルシートのzoomプロパティを使って、ズーム効果を実現するテクニックを紹介しよう。これは指定した要素の内側に含まれるすべての要素に対してズーム効果を適用できる便利なものだ。左はそのサンプルで、「zoom IN」や「zoom OUT」ボタンを押すと、黒い背景色の領域にある画像と文字が同時に拡大したり縮小したりする仕組みになっている。それではさっそくソースを見てみよう。 (大内 勇)

1

```
function zoomIn () {
  var Zoom = parseInt (zmArea.style.zoom) + 10 + '%';
  zmArea.style.zoom = Zoom;
}
function zoomOut () {
  var Zoom = parseInt (zmArea.style.zoom) - 10 + '%';
  zmArea.style.zoom = Zoom;
}
```

3

```
<IMG SRC="zmin.gif" onClick="zoomIn ()">
<IMG SRC="zmout.gif" onClick="zoomOut ()">
```

2

```
<DIV ID="zmArea" STYLE="width: 550px; height: 250px; zoom: 50%;">
<IMG SRC="winter.jpg" WIDTH="320" HEIGHT="240" ALIGN="left">
<H1 STYLE="font-size: 30pt">Snow World !!</H1>
</DIV>
```

POINT

ここで紹介するテクニックでは、スタイルシートの「zoom」プロパティとJavaScriptの「zoom」プロパティの2つを使っている。ソースのように簡単なのだが、注意してほしいポイントが1点だけあるので、最初に説明しておこう。スタイルシートの「zoom」は単独でも適用できるのに対し、JavaScriptの「zoom」はそれだけでは動作せず、操作する対象の要素にスタイルシートを必ず指定して、ページを読み込んだ時点でzoomの値をあらかじめ決めておかなければならないということだ。この点にだけ注意しておけば、特に悩むことなくこのテクニックを応用できるだろう。

まずはソース②を見てもらいたい。ID名「zmArea」(背景色が黒の領域)の大きさは550×250ピクセル、画像の大きさは320×240ピクセルを指定している。このことをふまえて左側のサンプルを見

てみよう。背景色が黒の領域に対して画像サイズの比率が明らかに小さくなっている。また「Snow World !!」の文字には30ポイントを指定しているのだが、画像と同様に小さくなっている。タイトルの「CSS Zooming」は24ポイントなので比較してほしい。zmAreaの領域にスタイルシートのzoomプロパティで50%を指定しているためだ。zoomはIE 5.5から使えるようになった新しいスタイルシートで、指定した要素の内側に含まれるすべての要素に対して有効になる。

次はダイナミックにズーム効果を出す、JavaScriptの「zoom」プロパティの説明をしよう。ソース①には「zoomIn」と「zoomOut」という2つの関数がある。その名のとおり、ズームイン(拡大)とズームアウト(縮小)の処理を行う関数だ。それぞれの関数の中にある「style.zoom」がズームを行うプ

ロパティで、その対象は「zmArea」(ソース①のID名「zmArea」)になる。今までのJavaScriptの感覚では、その要素自体の大きさが変わるようなイメージを持つかもしれないが、このzoomプロパティを使うと指定した要素は変わらずに、その内側だけ拡大、縮小するようになる。このサンプルでは拡大率と縮小率をそれぞれ10%になるように指定している。

ソース③はクリックするとズームインとズームアウトを行うボタンだ。onClickイベントで関数zoomInとzoomOutを呼び出しているだけだ。以上で指定領域をズームさせるTIPSの完成だ。冒頭で書いたとおりスタイルシートだけでも拡大や縮小表示はできるのだが、このTIPSのようにJavaScriptと連動させて、インタラクティブに使うと面白いだろう。

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

<SELECT>を制する

<SELECT>タグで作るプルダウンメニューは、フォーム内での選択肢として使われるだけでなく、JavaScriptと組み合わせてほかのページにジャンプする仕掛けに使えるなど応用範囲が広い。そのままでは地味な白や灰色のデザインのメニューだが、最新のブラウザを使えば、スタイルシートを適用してカラフルなものを作り込むことも可能だ。そこで今回は、プルダウンメニューにいろいろな飾りを付けてみる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きます。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“<SELECT>を制する”にチャレンジ!

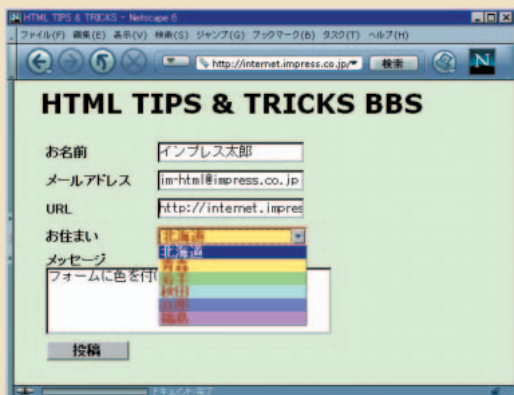
「HTMLパズルに挑戦しよう」宛先

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1月10日とさせていただきます。

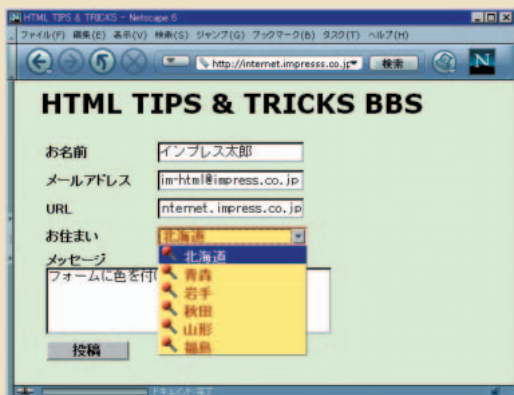
QUESTION 1 メニューを色分けせよ!



左のサンプルのようなプルダウンメニューは、ときどき目にすることがあるだろう。メニューの中の項目の背景がそれぞれ色分けされているものだ。また、文字の色も変更されている。こうしたテクニックを使えば、どのページでもかわりばえのしないプルダウンメニューにも、ちょっとしたアクセントを加えられる。スタイルシートがフォームの部品にも使えることはこの連載の読者ならご存知だろうが、<SELECT>にも適用できることには気づいていない人もいないのだろうか。そこで、ごく基本的な問題だがあえて出題してみた。この問題に一発で答えられなければ、TIPS & TRICKERとは言えないぞ。

ヒントを出すまでもないだろう.....

QUESTION 2 メニューに画像を入れる!



1問目は基本問題だが、2問目は真技の問題だ。左のサンプルを見てほしい。プルダウンメニューの各項目に、アイコン画像が付いている。ネットスケープ6やMozillaでは、スタイルシートを使ってこんなこともできるのだ。もちろん複雑なJavaScriptでメニューを作成しているわけではなく、ごく普通の<SELECT>タグを使っているだけだ。さて、いったいどうやるのだろうか。こんな技はどんなサイトや本でも紹介されていないし、さっぱり分からないという人は、Mozillaをインストールしたフォルダの下のどこかにある「test16.html」を探して見てみよう。Mozillaは今月も付録CD-ROMのBに収録してある。

「リスト」のためのスタイルシートを使う.....



[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