

HTML TIPS & TRICKS

第 52 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 / 佐藤和人

3月末にリリースされたインターネットエクスプローラ(IE)6のパブリックプレビューはすでに試してみただろうか。正式版リリースに先駆けて、これからはIE 6のTIPSを紹介していくのでぜひチェックしてほしい。また、このほかにもウェブページを効果的にアレンジするなど、今月はIE 5.5および6にターゲットを絞ってさまざまなテクニックをお届けしよう。



CD-ROM収録先 Magnavi Ip0106 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(4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



インターネットエクスプローラ6(パブリックプレビュー)



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



5月号「HTMLパズルに挑戦しよう」の解答

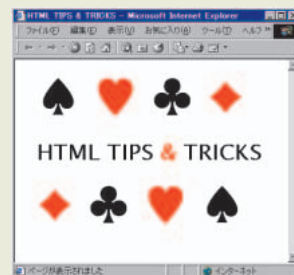
HTMLのごく基本的な機能を利用したもので、タグさえ使わない問題だが、いかがだったろうか。あまり意識されることはないが、ユニコードの採用はHTML 4の大きな特徴になっている。このHTML 4の新機能を使ってページにちょっとした遊びを加えてみたのがこのパズルだ。



ANSWER ① ハートマークを描け!

HTML 4の文字実体参照を使い、`♠`のように記述する。`&`や`©`と同じ要領だ。`♠`のように数値文字参照を用いてもよい。うまく表示されないときは、フォントの種類をいろいろ変えて試してみよう。

```
&spades;  
&hearts;  
&clubs;  
&diamonds;
```



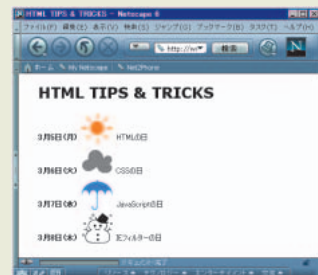
正解者：ENDEさん、Masahiko Murataさん、なるさん、鼎甲子さん、MIKAGE Shionさん、堀江さん、坂部和久さん、山口雅仁さん



ANSWER ② 雪だるまを描け!

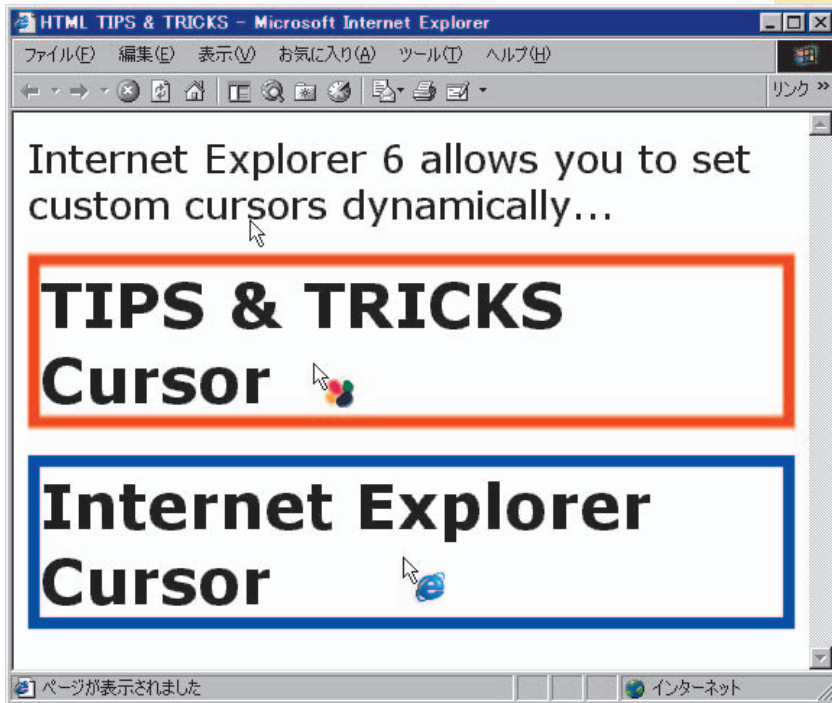
「`&#x`」から始まる数値文字参照を使い、ユニコードの2600から2603(16進数)の番号を直接指定する。HTMLファイルの文字コードをユニコードにして、文字を直接テキストとして埋め込んだ解答も正解とした。

```
&#x2600;  
&#x2601;  
&#x2602;  
&#x2603;
```



正解者：ENDEさん、Masahiko Murataさん、なるさん、鼎甲子さん、MIKAGE Shionさん、堀江さん、坂部和久さん、山口雅仁さん

自作のカーソルを表示する



IE 6パブリックプレビューの新機能をさっそく紹介しよう。スタイルシートにカーソル(マウスポインター)を変更する機能があるのは、この連載の読者ならご存じだろう。ページ上の好きな部分でカーソルを指の形にしたり、砂時計にしたりできる。しかし、今までは限られた種類のなかからカーソルを選べるだけで、好きな画像は割り当てられなかった。1999年6月号の「マウスカーソルを追いかけるアイコン」のように、スクリプトを書けば不可能ではないが面倒だ。IE 6では適切なファイルを用意すれば、短い記述でカーソルを置き換えられる。それでは試してみよう。(佐藤和人)



```
<P STYLE="cursor: url(tips.cur);">
```

Point

まずは、マイクロソフトのサイトから最新のベータ版である、IE 6パブリックプレビューをダウンロードしよう。

www.microsoft.com/japan/ie/preview/

IE 6ではパーソナルバーなどのブラウザのインターフェイスを中心に強化が図られているが、もちろんHTMLやスタイルシートにも新機能がいくつも追加されている。とくに、これまで中途半端だったHTML 4やCSS 1、2といった標準規格のサポートが進んでいる。ここで紹介する自作カーソルの指定も、CSS 2で定められているものだ。

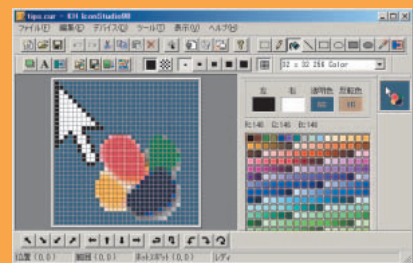
スタイルシートでカーソルを変える機能をおさらいしよう。次のように指定すれば、「div1」というクラス属性のDIVタグがあるところにマウスを動かしたときに、カーソルが砂時計の形に変わる。

```
Div.div1 { cursor: wait; }
```

このcursorプロパティに指定できる値には、waitのほか、pointer(指の形)やhelp(クエスチョンマーク)などがある。IE 6では、ここに背景画像と同じように「url」を使ってカーソルファイル(curファイル)を読み込めるのだ。

```
Div.div2 { cursor: url(mycursor.cur); }
```

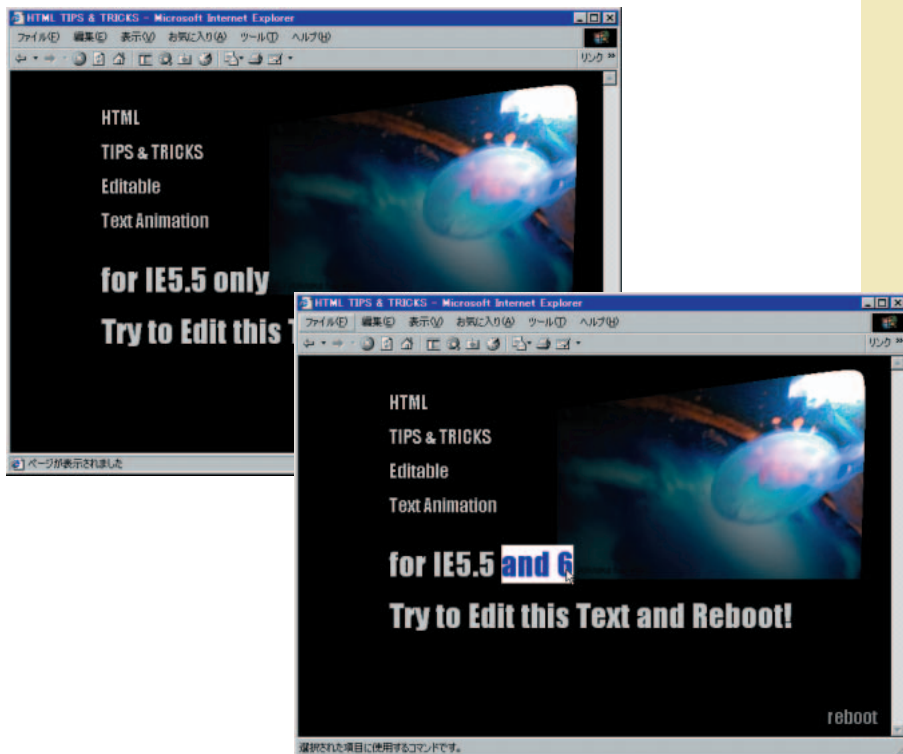
ウィンドズのカーソルファイルを作るには、専用のソフトが必要だ。普通のグラフィックソフトでは作成できない。たとえば、シェアウェア(1,000円)のKH IconStudio98はウィンドズ用のアイコン作成ソフトだが、2色からフルカラーまでのカーソルを作成する機能も備えている。



www.khsoft.gr.jp

IE 6のスタイルシートでは、curファイルだけでなくアニメーションカーソル(aniファイル)も指定できる。KH IconStudio98と同じ作者によるKH ANIMedit97(シェアウェア、1,200円)を使えば、カーソルがアニメーションするウェブページも作れるようになるぞ。

テキストを書き換えられるようにする



ホームページを訪れた人が画面のなかで文字を書き込めるところといえば、フォームのテキストボックスくらいしかない。ところがIE5.5以降では、フォーム以外の通常の文章も直接編集できるようになっている。とはいえ、サーバーにアップロードされているファイルを書き換えるわけではなく、あくまでブラウザーに表示されたウェブページのテキストを書き換えるだけなので、使い道を思いつくのは大変だ。そこで、今回は演出に趣向を凝らして、ちょっとしたお楽しみサンプルを作成した。左のテキストをいろいろ書き換えて、ウィンドウ右下の「reboot」ボタンをクリックしてみよう。（藤井幸孝）



```
<DIV ID="text0" contenteditable="true">.....</DIV>
```

POINT

編集できるテキストを用意するためのコードは、上に書いたように驚くほど簡単だ。<DIV>タグやタグなどに、IE 5.5で追加された新しいプロパティ「contenteditable="true"」を付けておくだけでよい。

```
<DIV ID="text0" contenteditable="true">
.....(書き換え可能にしておきたいテキスト)
</DIV>
```

たったこれだけで、IE 5.5でこのページを見れば、<DIV>タグで囲まれた部分のテキストを、好きなように書き換えられる。サンプルを使って試してほしい。マウスでポイントした部分の文字を消したり書き換えたりできただろうか。

今回は、以前に紹介したDOMを使ったテキストア

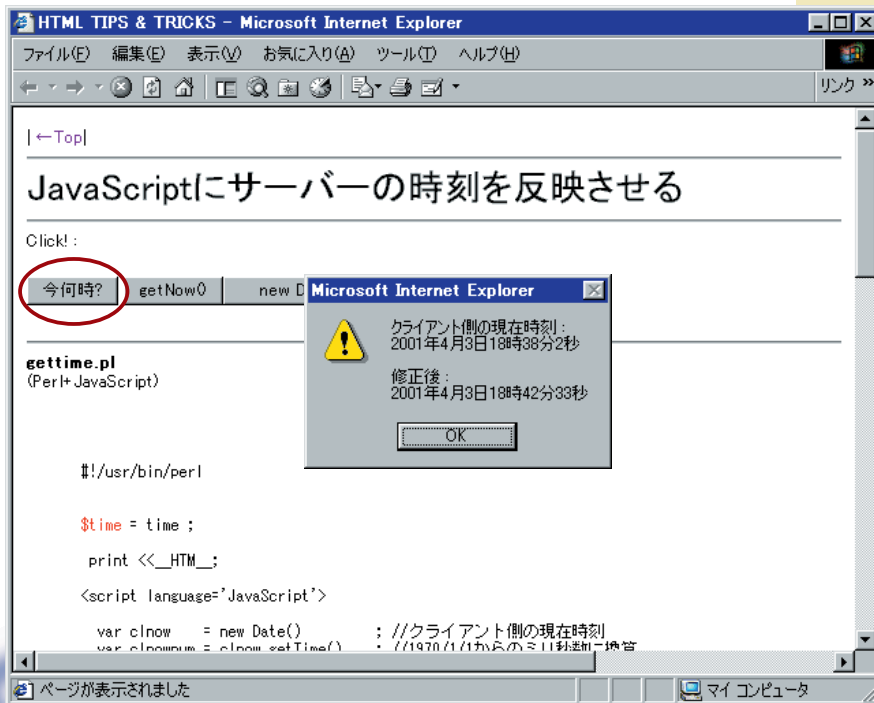
ニメーションと、先月号の光源フィルター(スポットライト)を組み合わせ、簡単なVJ作品のようなサンプルを作ってみた。映像に合わせて表示されるテキストがすでに設定されたものだけでなく、自由自在に書き換えられるように、フェードイン表示されるアニメーションの部分にcontenteditableを適用してみた。サンプルの6行の文字列はすべて「editable(編集可能)」になっているので、これを好きな単語やメッセージに書き換えて、ウィンドウ右下にある「reboot」をクリックすれば、再表示するときに書き換えたメッセージが表示される。もちろん日本語もOKだ。なお、「更新」ボタンで表示をリフレッシュすると、書き換えたテキストが元に戻ってしまうので必ず「reboot」を使うようにしましょう。

さらに各種のスク립トやSVGを駆使すれば、マウスの動きで映像やテキストを操作できる本格的

なページも作れるはずだ。ホームページを見に来てくれた人にこれで遊んでもらおう。

書式そのものが単純なだけに、このTIPSで一冊難しいのは、その使い道だろう。ページの中身を書き換えて表示するのは作者にとってうれしいことではないし、contenteditableを使って書き換えられたテキストは、フォームに入力した文字のように送信できない。いったんブラウザーを閉じてしまうと、書き換えた文字列は失われてしまう。そこで、挨拶状や定型文などのひな型を表示するサイトで使ってみてはどうだろう。「editable」にしたクールなテンプレートを公開しておけば、宛名やメッセージ部分を書き換えたあとにブラウザーの印刷機能でプリントアウトして使える。フォームを使ったときに比べてテキストボックスの枠線がじゃまをしないので、きれいなフォーマットが作れる。

サーバーの時刻に合わせる



JavaScriptのDate()は時間を扱う関数で、ちょっと応用すればホームページの端に「今月のカレンダー」のような飾りを自動的に作れる便利なものである。しかしこれには使用上、ちょっと困った点がある。たとえば、アクセスしたマシンの日付が1999年5月に設定されていると、カレンダーを作ってもその1999年5月が「今月」と表示されてしまう。月単位ならばともかく、分単位で何らかの処理を行うのには向いていない。そこで、サーバーの時刻をJavaScriptに取り込んで、アクセスしてきた環境によらずサーバー側の時刻を常に表示して利用するテクニックを紹介する。
(高橋登史朗)



1

```
<!--#exec cmd="./time.pl"-->
<FORM>
<INPUT TYPE="button" VALUE="今何時?"
onclick="alert('修正前:'+new Date()+'\n\n'+修正後:'+new Date(getNow()))">
```

2

```
$time = time ;
print <<_HTM__;
<SCRIPT LANGUAGE='JavaScript'>
var cnow = new Date() ;
var cnownum = cnow.getTime() ;
var svnownum = $time * 1000 ;
var sa = svnownum - cnownum ;
function getNow(){
var now = new Date() ;
return now.getTime() + sa ;
}
</SCRIPT>
__HTM__
```

POINT

今回のTIPSはサーバー側でPerlとSSIを準備しておき、クライアント側でJavaScriptを動作させる必要がある。まずは先にソース①を説明しておこう。これはPerlスクリプトでJavaScriptを書き出すのがおまな役割で、「test.pl」として実行可能な状態でアップロードしておこう。ここではまず

```
$time = time ;
```

として、サーバーの時刻を\$timeというPerlの変数にセットしている。UNIXやウィンドウズのサーバーならば1970年1月1日から現在までの秒数が取まる。JavaScriptのget Time()は1970年1月1日から現在までのミリ秒を返すので、ちょうど1000分の1の値を返すわけだ。次の

```
print <<_HTM__;
```

```
.....(ここにHTMLを記述する)
__HTM__
```

は、PerlによるHTMLの書き出し部分だ。ここでは<SCRIPT>タグごとJavaScriptを記述する。上記の\$timeをJavaScriptに埋め込むのには、

```
var svnownum = $time * 1000
```

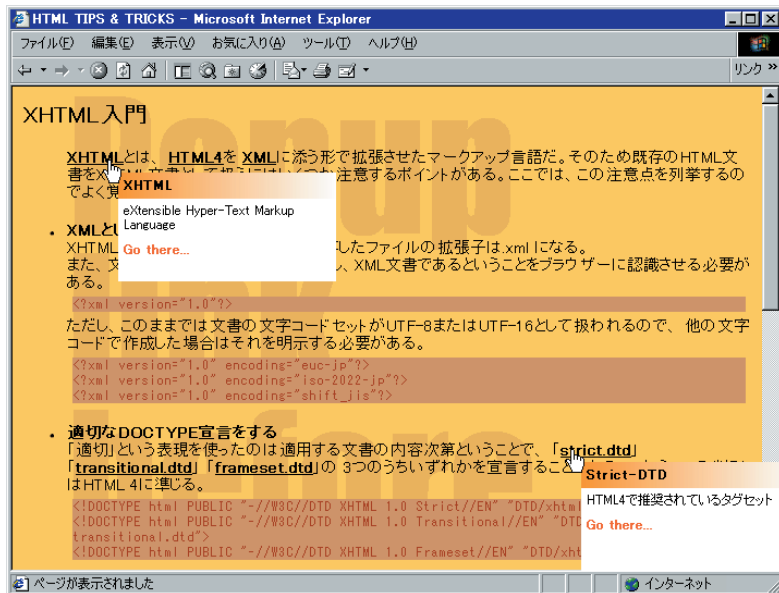
と記述する。ここでJavaScriptの変数svnownumへPerlの変数\$timeを1000倍して代入している。あとは、あらかじめ得ておいたクライアント側の時刻とsvnownumとの差を引算して、new Date() するごとにその差を修正してやればよいわけだ。そしてHTML本体にはソース②の

```
<!--#exec cmd="./test.pl"-->
```

のように記述すると、PerlスクリプトがSSIによって呼び出され、結果がJavaScriptの形でHTMLファイルに埋め込まれる。最終的にはJavaScriptがユーザー側のマシンで実行されてもサーバー側の時刻を反映させた正しい時刻が使われるというわけだ。CD-ROMではPHPでの例も示したので、PHP派はこちらを参照して欲しい。

今回はやや異質なTIPSかもしれないが、いずれにせよJavaScriptのDate()関数にはローカル側の時刻を素直に取り込んで使ってしまう欠点があることを理解しておこう。曜日や日付、時間帯によって異なるコンテンツを表示させるのにこの関数を使っている人は、間違った時刻を表示するクライアントでどう見えるのかということにも注意してもらいたい。

リンク先の説明をポップアップさせる



ここで紹介するTIPSは、リンク先の簡単な説明を表示させてからジャンプさせるというものだ。ステータスバーにリンク先の情報を表示させてからジャンプするというテクニックがあるが、これも今ではもう古くさく、ステータスバーの変化に気付きにくい。そこで今回は左のサンプルのように、リンク先の情報をポップアップウィンドウではっきりと表示させる。ウィンドウの「Go there」をクリックすると指定したページにジャンプする。ポップアップウィンドウの枠やテキストの色はカスタマイズでき、背景にグラデーションをかけるのも簡単なので、ちょっとした解説を加えるのに使ってみよう。(大内 勇)



1

```
<B onclick="popupProp('XHTML','eXtensible Hyper-Text Markup Language','http://www.w3.org/TR/xhtml1/','100')">XHTML</B>
```

2

```
<DIV ID="popupWin" STYLE="display:none">
<DIV ID="Title"></DIV>
<DIV ID="Text"></DIV>
<DIV ID="Link"></DIV>
<DIV onclick="parent.location.href=Link.innerText">Go there...</DIV>
</DIV>
```

3

```
<SCRIPT LANGUAGE="JScript">
var popup = window.createPopup();
function popupProp(ttl,info,link,h) {
  Title.innerText = ttl;
  Text.innerText = info;
  Link.innerText = link;
  goPop(h);
}
function goPop(h) {
  var popupBody = popup.document.body;
  var posX = event.x + 10;
  var posY = event.y + 10;
  popupBody.style.backgroundColor = "#ffffff";
  popupBody.innerHTML = popupWin.innerHTML;
  popup.show(posX,posY,200,h,document.body);
}
</SCRIPT>
```

POINT

ソース①はポップアップウィンドウを開くための文字に指定するタグだ。onClick イベントで関数「popupProp」を呼び出すと、マウスをクリックしたときにポップアップウィンドウを表示させるようになっていたのだが、そのときに4つの値を渡している。括弧内のシングルクォートで囲った部分は、前から順に

(「タイトル」、「リンク先の簡単な説明」、「URL」、「ポップアップウィンドウの高さ」)

となる。

ソース②は少し複雑な役割を果たしている。この部分はソース①からソース③に渡された値をいったん格納するために必要だ。ここでは最初に、ソース③の全体にpopupWinというID名を指定し、

スタイルシートで非表示(display:none)にしておく。ID名Title,Text,Linkの各DIVタグは、ソース③から渡される文字列の受け皿になり、最後のDIVタグは、ソース①で指定したURLへジャンプさせるためのリンクとなる。

最後のソース④がポップアップウィンドウに関する各種の処理を行うスクリプトの部分だ。最初の行の「createPopup()」がポップアップウィンドウを作るメソッドだ。関数popupPropはソース①から呼び出されるもので、この中では「innerText」を使ってソース①から渡された値を、ソース③の受け皿(ID名Title,Text,Linkの各DIVタグ)に入れるようにしている。その後で関数「goPop」に引数h(ポップアップウィンドウの高さ)を渡して処理を移している。関数goPopではマウスの位置に合わせたポップアップウィンドウの表示位置(posX, pos

Y)や背景色を指定し、その次でソース⑤で指定したID名popupWin内の各要素をポップアップウィンドウ(popupBody)の要素に置き換える。最後にshow()でポップアップウィンドウを表示状態にさせている。なお、カッコ内は先頭から順に「X座標、Y座標、幅、高さ、基点」を指定してゆく。ちなみに今回の場合は基点に「document.body」を指定しているため、ブラウザウィンドウ内の左上位置がX座標=0、Y座標=0になる。

参考までに「innerText」と「innerHTML」の違いを書いておく。「innerText」は指定したオブジェクトの内側に含まれるテキストのみが該当し、「innerHTML」は指定したオブジェクト内側のタグを含んだ全要素を含むものになっている。今回のTIPSではこれら2つの値が両方とも登場しているので、混同しないようにしてほしい。

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

・IE 6を制する・

IE 6で追加された新機能の中には、地味だが重要な意味を持つものが多い。まだ完全とは言えないものの、これまでサポートされなかったHTML 4やCSSの細かい部分が拾い上げられている。たとえば、見た目に変化はないが、<TH>と<TD>タグでHEADERS、SCOPE、ABBR、AXISといったアクセシビリティ関連の属性が正式に採用されている。そこで今月は、IE 6がW3Cの規格をサポートしているかどうかを実験す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"IE 6を制する"にチャレンジ！

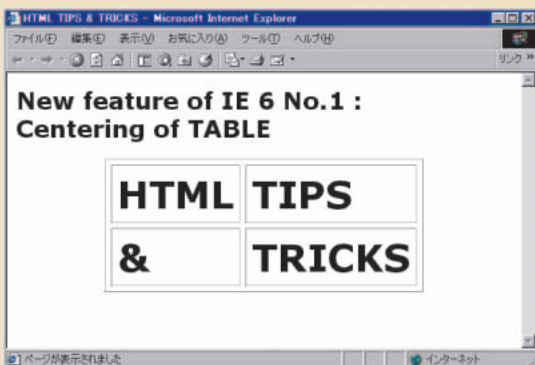
「HTMLパズルに挑戦しよう」 宛て先

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5月10日とさせていただきます。

QUESTION 1 centerなしでセンタリング！

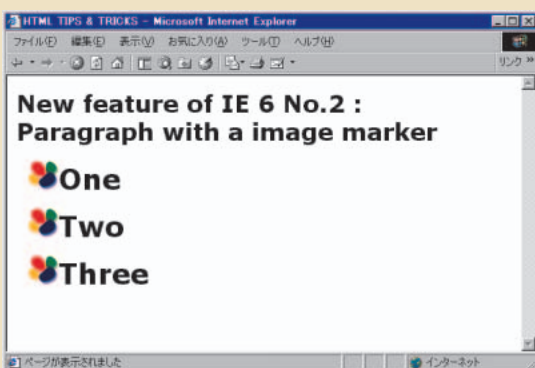


第1問は「テーブルをセンタリングする」というだけの問題だ。HTML入門書の最初に出てきそうな単純な問題で、がっかりしたかもしれない。しかし、この問題はなかなか奥が深いのだ。条件の1つとして、<CENTER>タグを使わずにスタイルシートで実現してほしい。もう1つの条件として、text-align プロパティを使わずにセンタリング表示してほしい。実は、<TABLE>タグにtext-align: centerを指定するとセンタリングされてしまうのは、CSSの文法としては間違っているのだ。これまで以上に厳密に標準規格をサポートしているIE 6では、こうした間違いが修正されている。さあ、いったいどうすればいいのだろうか。



<!DOCTYPE ~>を記述する.....

QUESTION 2 自動的に画像を入れる！



第2問は、CSSを利用したちょっとしたトリックだ。左のサンプルの「One」「Two」「Three」という部分は<P>タグで作った段落だ。各段落の前に画像が置かれているが、これはタグで指定しているのではなく、スタイルシートの機能を使っている。CSS 2のサポートが進んでいるネットスケープ6やMozillaでは、content プロパティを使って要素の前に自動的に文字や画像を入れることができるが(2000年9月号参照)、IE 6はまだそこまで進んでいない。CSS 1だけを使って段落の前に画像を置くのが条件だ。もちろん背景画像を使うのではない。CSS 1で背景以外に画像を扱える機能があるかどうかをじっくり調べてみよう。



displayを変える.....



[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