

HTML TIPS & TRICKS

第56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

1997年2月号に始まったこの連載も56回目です。ついに最終回を迎えることになった。ブラウザの進化に合わせてこれまで300以上ものTIPSを紹介してきたが、これらがウェブでの表現の幅を広げてきたと自負している。今回はXMLを利用したTIPSなどを紹介するが、これらを今後のウェブの方向を考えるうえでの参考にしてもらえれば幸いである。



CD-ROM収録先 Magnavi Ip0110 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(7月29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(パワリックプレビュー)



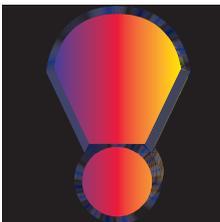
ネットスケープナビゲーター4以上



ネットスケープ6以上



Mozilla



9月号「HTMLパズルに挑戦しよう」の解答

今回はひさしぶりに多くの方から応募をいただいた。W3Cのサイトに注目していれば、第1問のようにブラウザの最新機能が発見できることもある。第2問では、<RT>タグを列挙して1つずつ表示を切り替えるスクリプトは、残念ながら不正解とさせていただきます。



ANSWER 1 かっこを使いこなせ!

<RUBY> ~ </RUBY>の間で、かっこを<RP> ~ </RP>で囲む。ルビに対応したブラウザでは<RP>の内容は表示されない。マイクロソフトのサイトには<RP>タグが記載されていないが、IE 5以上でサポートされているようだ。

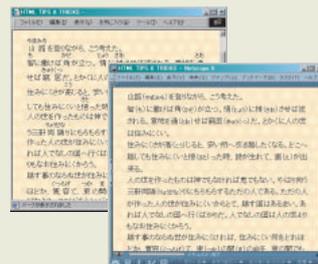
```
<RUBY><RB>山路</RB><RP> ( </RP><RT>やまみち</RT><RP> ) </RP></RUBY>
```



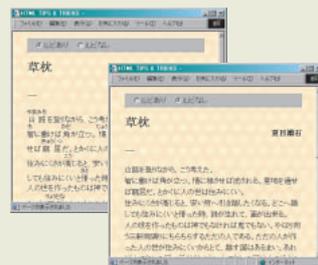
ANSWER 2 ルビの有無を切り替える!

このページでメインとなるスタイルシートとは別に、RT { display: none; }だけの<STYLE>タグを追加する。スクリプトで配列styleSheetsを使ってこの<STYLE>タグの有効と無効を切り替えれば、ルビを一度に表示したり消したりできる。

```
<STYLE TYPE="text/css" DISABLED>
RT { display: none; }
</STYLE>
<SCRIPT TYPE="text/javascript">
function rubyswitch(on) {
  document.styleSheets[1].disabled = on ? true : false;
}
</SCRIPT>
```

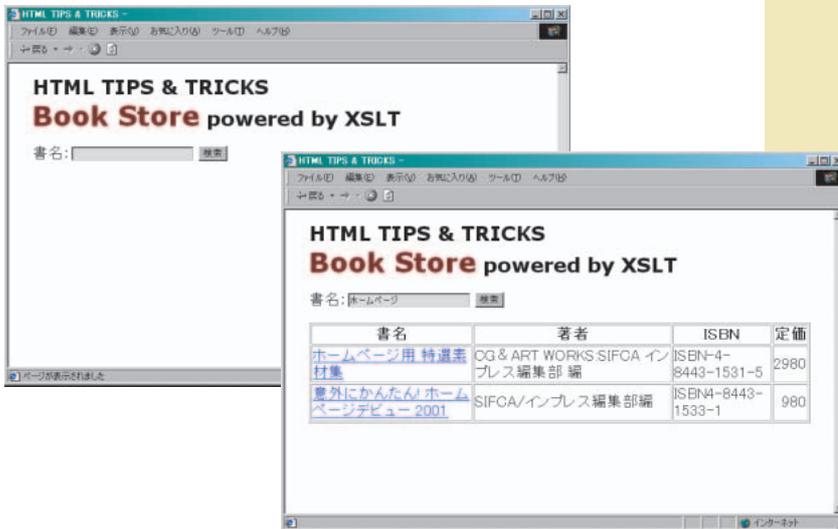


正解者: Kazuhisa Sakabe, 山口雅仁, Masahiko Murata, ENDE, 岩田裕之, 堀江、がず、よしとも、なるる、鼎甲子、うおまさ@home (敬称略)



正解者: Kazuhisa Sakabe, Masahiko Murata, ENDE, 堀江、よしとも、なるる (敬称略)

XSLTでXMLデータを検索する



1

```
<XML ID="bookdata" SRC="book.xml"></XML>
<XML ID="bookxsl" SRC="book.xsl"></XML>
```

2

```
<SCRIPT TYPE="text/javascript">
function search (keyword) {
  node = bookxsl.XMLDocument.selectSingleNode("//xsl:for-each");
  node.setAttribute("select", "//book[contains(title, '" + keyword + "')");
  str = bookdata.XMLDocument.transformNode(bookxsl.XMLDocument);
  view1.innerHTML = str;
}
</SCRIPT>
```

連載の最終回にふさわしく、未来のファイル形式XMLを使ったTIPSを紹介しよう。左のサンプルは一見CGIを使った検索フォームのようだが、実はJavaScriptで外部のXMLファイルを読み込んで、ブラウザ上でデータを検索しているのだ。これまで外部のデータを読み込むにはインラインフレームを使うしか手がなかったが、XMLを使えばHTMLとは別にデータを保存して、JavaScriptから読み込んだり検索したりできる。しかも最近流行のXSLTを使ってXMLデータをHTMLのテーブルやリストに変換することも自由自在だ。この仕組みがわかれば新時代のウェブ技術もばっちりだ。(佐藤和人)

Point

XSLT(XSL Transformation)とは、XML形式のスタイルシート言語XSL(Extensible Stylesheet Language)の機能の1つで、あるXMLデータを別のXMLデータに変換するものだ。XSLTを使うと、XMLデータをHTML形式に変換してウェブブラウザで自由に表示できるようになるため、現在大きな注目を集めている。

まずソース①のように、IEの独自タグ<XML>を使って外部XMLファイルを読み込む。book.xmlは書籍データを記述したXMLファイルで、book.xslはそのXMLデータをHTML形式に変換するためのXSLTを記述したXSLファイルだ。XSLもXML形式のため、<XML>タグで読み込める。book.xmlとbook.xslの内容についてここでくわしい説明はしない。付録CD-ROMに収録したファイルを見ながら、マイクロソフトの次のページを参照してほしい。

① msdn.microsoft.com/library/en-us/xmlsdk30/htm/xmmscxmlreference.asp

② msdn.microsoft.com/library/en-us/xmlsdk30/htm/xmrefxsltreference.asp

ソース②の「search」はXMLデータを検索して表示させる関数だ。book.xslファイルのなかに次のような行がある。XMLデータの「<book>要素をすべて取り出せ」という意味だ。

```
<xsl:for-each select="//book">
```

関数searchの最初の2行は、この<xsl:for-each>要素を入力されたキーワードに従って変化させるものだ。selectSingleNodeメソッドで<xsl:for-each>要素にアクセスし、setAttributeメソッドでselect属性を変化させる。その結果、<xsl:

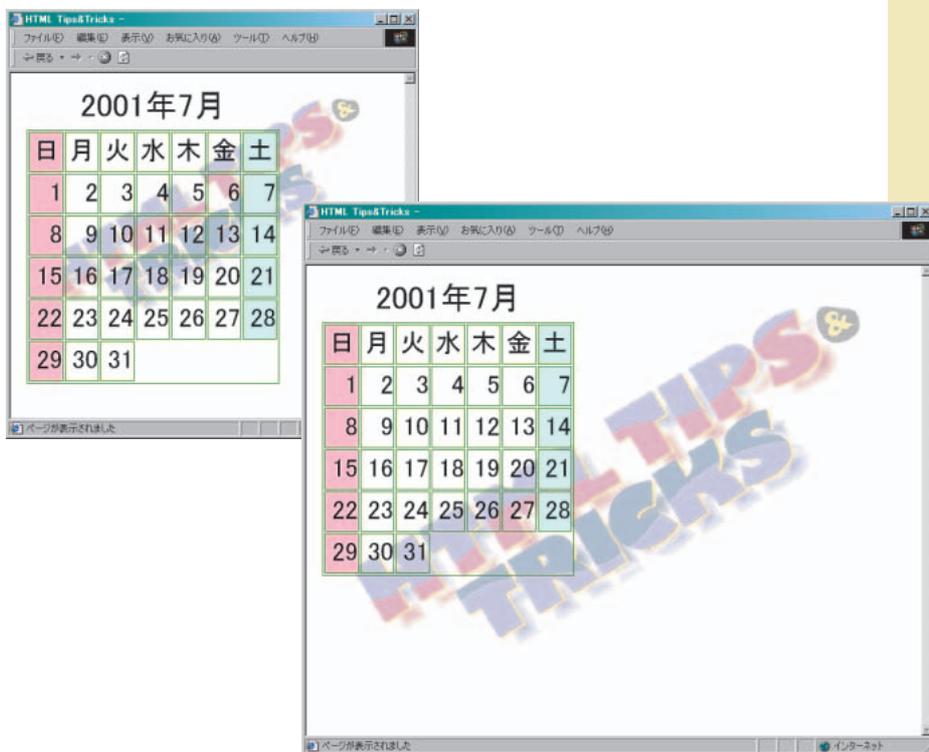
for-each>要素が次のように変わる。

```
<xsl:for-each select="//book[contains(title, 'キーワード')]">
```

これにより、<xsl:for-each>要素の意味が「<book>要素の下の<title>要素にキーワードが含まれるものを取り出せ」になる。

次に、transformNodeメソッドでXMLデータとXSLTを組み合わせると、キーワードに一致したデータがHTML形式の文字列になって取り出せる。この文字列をページのなかに表示させればいい。今回のサンプルは残念ながらIE 6でしか動作しない。付録CD-ROMにはIE 5や5.5でも動くサンプルを収録してある。検索機能はなく、XMLデータを読み込むだけのものだが、XSLTの力実感できるだろう。

背景画像をウィンドウサイズに合わせる



ホームページで大きい背景画像を使うときに、ウィンドウサイズと背景画像のサイズを合わせることができれば、きれいなデザインになるのと思ったことはあるだろう。そんなときは、ここで紹介するTIPSを使ってもらいたい。左はこのサンプルだが、ウィンドウサイズを変えると同時に背景画像のサイズも変わっていることがわかるだろう。ただし、このTIPSには弱点もある。上下のスクロールが必要なページの場合、スクロールした部分の背景がなくなるといったことだ。そのため、このTIPSは画面におさまるページで使うようにしてもらいたい。ソースはとても簡単だ。興味のある方はさっそくトライしてみよう。

(大内 勇)



```
<IMG src="title.gif" width="100%" style="position:absolute;top:0;left:0;z-index:0">
<DIV style="position:absolute;top:0;left:0;z-index:1">
  (ここにページの内容を記述)
</DIV>
```

POINT

背景画像というと、<BODY>タグの[background]属性を使って指定する手法が普通だがここで紹介するTIPSは上記のソースのとおり<BODY>タグを使うことなく背景画像を表示させる。では、どのような手法を使っているのだろうか？それはスタイルシートのレイヤー機能を使っているのだ。

まず、最初に基本的な構造を説明しておく。ソースを見るとタグと<DIV>タグを使っていることがわかるだろう。このソースは<BODY>から</BODY>までの間に記述し、ページの構成要素はソースの<DIV>から</DIV>までの内側に記述する。これでタグと<DIV>タグが別のレイヤーになるため、タグで指定した画像を背景にできるのだ。

次はこのTIPSの仕組みを説明しよう。タグと<DIV>タグには同じようなスタイルシートを指

定している。それぞれ[position,left,top]属性で同じ位置に指定し、[z-index]属性を使って重ね合わせの順番(値が大きいほうが上に重なる)を指定している。これでレイヤーが同じ位置に重なり合うようになるため、[z-index]の値が小さいタグが背景画像になるのだ。背景画像がウィンドウのサイズに合う仕組みは、タグの[width]属性を100%に指定することで実現している。本来なら背景画像をウィンドウサイズに合わせるのだから、[width]属性に加えて[height]属性の値も100%にしなければいけないのだが、これを指定してしまうと画像の縦横比が崩れてしまい、ウィンドウサイズによっては非常に格好悪くなってしまふ。[width]属性だけを指定しておけば、画像サイズの縦横比が維持されるため崩れない。このTIPSでは使っていないが、次のようなスタイルシートを<DIV>タグに指定すると、フォントサイズ

もウィンドウサイズに応じて変えることができるため、さらにCoolなページになるだろう。

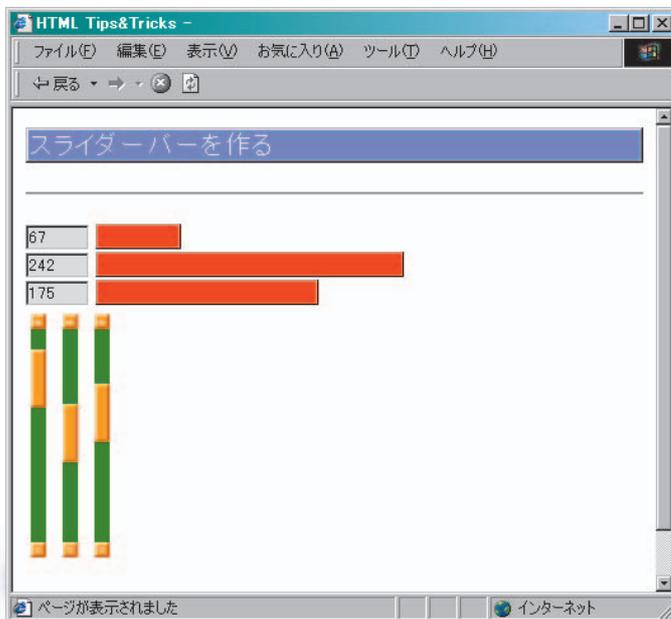
```
font-size:expression(body.clientWidth/10)
```

ただし、この方法はIE5以降でしか使えない。注意してほしい。

このページの右上の囲みでも触れたが、このTIPSを上下スクロールが必要なページに組み込むと、スクロールした部分の背景画像が表示されなくなってしまうため、なるべくスクロールする必要のないページで使おう。

なお今回のTIPSとは関係ないが、サンプル画面の上に重なっているレイヤーで使っているカレンダーはJavaScriptで作られている。誌面の都合上これに関しては説明しないが、興味がある人は付録CD-ROMのソースを参考にしてほしい。

スライダーバーを作る



1

```
.sliders {
  font-size:614px;
  width:20px; height:200px;
  border :4px solid #ffffff}
```

2

```
<!--グラフ-->
<input type="text" id="inputdata1" style="width:50px">
<span id="g1" class="gbar"></span>
```

3

```
<!--スライダー-->
<textarea
  id = "data1"
  class = "sliders"
  onscroll = "inputdata1.value=this.scrollTop;
  document.all('g1').style.width=this.scrollTop">
</textarea>
```

POINT

まずソース①を見てみよう。これが今回のスライダーバー本体だ。え？ただのテキストエリアじゃないかって？その通り。このテキストエリアに魔法のCSSをかぶせてスライダーに仕立て上げてみた、というわけだ。そこで、このテキストエリアを調べる前に、class属性で指定しているCSSソース①を見て欲しい。まず、

```
width:20px ;
height:200px ;
```

が第1のポイントだ。これで、テキストエリアの幅が20px、高さ200pxに絞込まれる。これがスクロールバーのエリアになるわけだ。もし、この幅と高さを逆にすれば、横向きのスクロールバーエリアが確保されることになる。でも、実はこれだけでは、スクロールバーは現れてくれない。

ご存じのように、テキストエリアにスクロールバーが現れるためには、「テキストエリアから文字が溢れていて、スクロールしないと全部を見れない」という状態になる必要があるのだ。そこでまず、下のように、

```
font-size:614px ;
```

とフォントサイズを大きくする。これで文字が溢れて、スクロールバーが現れる。バーの太さを変更したい時は、

```
border :4px #ffffff ;
```

のpx値を変更すると良い。これを小さくするほど太くなり、大きくするほど細くなる。#ffffffの部分の色はページの背景色に合わせてよう。サンプル

は作った時のスライダーの可変数値を500に微調整しているが、これは、font-size:614pxとborder:4pxの組み合わせで指定する。仮に700に指定したければ、

```
font-size:703px ;
border :4px;
```

となる。この辺は、関数を別で作っておくと便利かもしれない。

最後に、ソース③のonscroll内のスクリプトだが、これは、スライダーバーをドラッグした値を表示し、グラフ化するためのものだ。もちろん使い方はこれに限らない。可変数値の入力が必要な場所なら、気軽に使える。



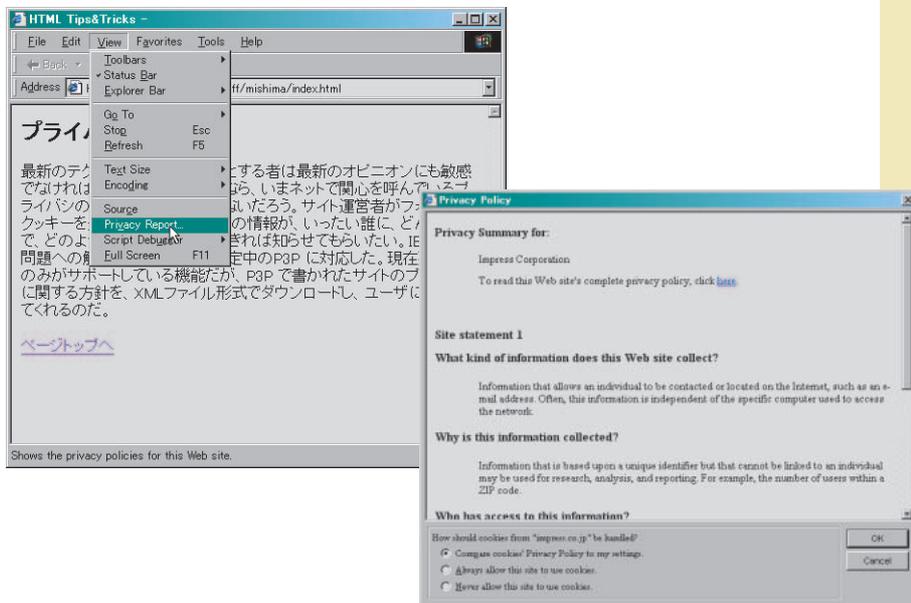
今回は最終回にふさわしいITIPSということで、少しトリッキーな技を紹介しよう。見た目は普通のスライダーバーだが、普通と違うのは、実は、HTMLの<TEXTAREA>のスクロールバーを利用して作っているところだ。スライダーをドラッグして動かすことで、いろいろな可変数値入力のコントロールとして使える。ただ欠点として、トリッキーなTIPSだけあって利用できる環境がウィンドウズのIE5以上に限定されてしまうが、スクリプト部分はいたってシンプルで、簡単にできる。クロスブラウザなDHTMLを使うまでもない限定された場所などで気軽に使えて便利だ。

(高橋登史朗)

プライバシーポリシーを表明する



6



最新のテクノロジーを使いこなそうとする者は、最新のオピニオンにも敏感でなければならない。そう思う君なら、いまネットで関心を呼んでいるプライバシーの問題にも無関心ではないだろう。そこで今回はW3Cが策定中のP3Pを利用して、サイト運営者がフォームやクッキーを通じて集めるユーザーの情報を、いったい誰が、どんな目的で、どのように使うのかを表明するTIPSを紹介しよう。残念ながら、現在のところ対応ブラウザがIE6の英語版だけしかないのだが、日本の製品版では搭載される可能性の高い機能だけに、学んでおく価値があるだろう。

(藤井幸孝)



1

```
<HTML><HEAD>
<link rel="P3Pv1" href="p3p.xml">
</HEAD></HTML>
```

2

```
<META xmlns="http://www.w3.org/2000/12/P3Pv1">
<POLICY-REFERENCES><EXPIRY max-age="172800"/>
<POLICY-REF about="/P3P/PolicyMain.xml">
<INCLUDE/*></INCLUDE>
<EXCLUDE>/cgi-bin/*</EXCLUDE>
</POLICY-REF>
</POLICY-REFERENCES>
</META>
```

3

```
<POLICY xmlns="http://www.w3.org/2000/12/P3Pv1"
discuri="http://www.blueyonderairlines.com/ourprivacypolicy.html">
<ENTITY><DATA-GROUP> ( サイト運営者の情報を書く ) </DATA-GROUP></ENTITY>
<ACCESS><contact-and-other/></ACCESS>
<STATEMENT>
<CONSEQUENCE>関心を持たれた商品情報をお送りします。</CONSEQUENCE>
<PURPOSE><pseudo-analysis/></PURPOSE>
<RECIPIENT><other-recipient/></RECIPIENT>
<RETENTION><business-practices/></RETENTION>
<DATA-GROUP><DATA ref="#user.home-info.email">
<CATEGORIES><online/></CATEGORIES>
</DATA></DATA-GROUP></STATEMENT>
</POLICY>
```

POINT

英語版のIE6では、メニューバーの[View]のなかに[Privacy Report]という項目があり、そのサイトの持つプライバシーポリシーを表示できる。この機能で利用されるP3Pのポリシーは、ポリシー参照ファイルとポリシーファイルの2つから構成され、それぞれはXMLで記述される。ソース②の参照ファイルは、ポリシーファイルの格納場所を示している。<EXPIRY>タグはポリシーの寿命で、max-age="mm"で「mm秒間有効」という意味だ。<POLICY-REF about="filename.xml">はポリシー本体を相対URLで表している。<INCLUDE>と<EXCLUDE>はポリシーの適用/非適用範囲をディレクトリー単位で表す。例では/cgi-bin ディレクトリー以外のすべてのディレクトリーが適用対象となる。そしてポリシーの本体が、ソース③のポリシーファイルだ。まずxmlnsでP3PのDTDを示した後、必

須項目のdiscuriで、ポリシーをくわしく説明したページのURLを挙げる。<ENTITY><DATAGROUP>はサイト運営者の情報、<STATEMENT>が実際のポリシー部分になる。個人情報収集の目的(PURPOSE)、受取人(RECIPIENT)、保管ポリシー(RETENTION)、対象データ(DATA-GROUP)のほか、オプションで言葉による説明(CONSEQUENCE)を書くこともできる。<DATA GROUP>には複数の<DATA>とその<CATEGORIES>を記述する。ここでは「匿名分析(pseudo-analysis)のために、サイト運営者以外(other-recipient)がネット(online)情報であるメールアドレス(ref="#user.home-info.email")を集め、ビジネス判断で保存する」ことを知らせている。<ACCESS>タグは、サービス提供者に質問や不満があれば苦情を出せるかどうかを示すためのものだ。contact-and-otherは、連絡用情報があ

り、それを閲覧できるという意味になる。これらのタグや値はごく一部にすぎない。よりくわしく知りたい人は、W3CのP3P仕様書④Jump01または日本語訳④Jump02を参考してほしい。さて、気になるポリシーの提供方法だが、まずはポリシー参照ファイルをブラウザに渡してやらなければならない。W3Cでは、サイトの特定の場所「http://ドメイン名/w3c/p3p.xml」に置く、もしくはHTTPヘッダーに記述する、たとえば「P3P: policyref="http://www.impress.co.jp/P3P/p3p.xml"」のような方法、HTMLのLINKタグを使う「<link rel="P3Pv1" href="p3p.xml">」の3種類の方法を定義している。今回の例ではソース①のように<LINK>タグを利用してみた。

④Jump01 www.w3.org/TR/P3P/

④Jump02 mda.or.jp/enc/w3c/cr-p3p-20001215j.html

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

3つのブラウザを制する。

これまで56回にわたってさまざまなTIPSとパズルをお届けしてきたが、新しいブラウザの登場や気付かなかったテクニックの発見によって、今では古くさく見えるサンプルもある。そこでこれまでの連載の復習として、以前にこの連載で紹介したダイナミックHTMLを、IE 4~6、ネットスケープナビゲーター4、ネットスケープ6とMozillaという3つのシステムのブラウザで同じように動作するように書き直すパズルに挑戦していただく。正解者には抽選で折り畳み傘をプレゼントさせてもらおう。なお、正解は9月29日にウェブページ内 [Jump internet.impress.co.jp](http://internet.impress.co.jp) で発表する。それでは頭をやわらかくして、今月のテーマ"3つのブラウザを制する"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、次の宛て先にメールしよう。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9月10日とさせていただきます。

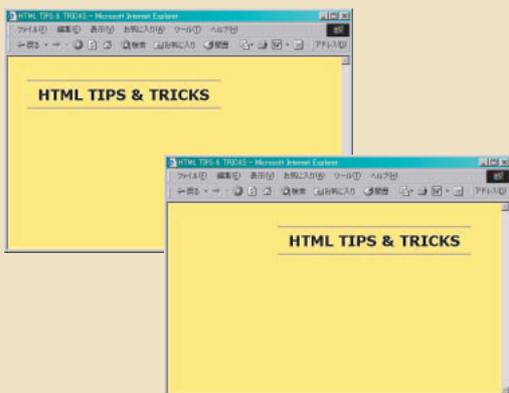
QUESTION 1 マウスオンでメッセージを表示せよ！



右のサンプルは、1999年3月号(単行本「ホームページ裏ワザ大全 HTML TIPS & TRICKS」では99ページ)で紹介した「2つのブラウザでダイナミックHTMLを動かす」と同じく、マウスをリンクの上に載せるとそれぞれ違ったメッセージを表示させるページだ。これをナビゲーター4とIEだけでなく、ネットスケープ6とMozillaでも動作するように改良するのが第1問だ。さらに、1999年3月号のようにnavigatorオブジェクトを使ってブラウザを判別せずに、ほかの方法を使ってダイナミックHTMLが使えるかどうかを調べることを条件とする。その判別方法も、これまでの連載をよく見れば実は何度か紹介されている。

ヒント getElementByIdでオブジェクトを取得して.....

QUESTION 2 オブジェクトを自由に動かせ！



第2問も第1問と同じように、1999年4月号(単行本「ホームページ裏ワザ大全 HTML TIPS & TRICKS」では100ページ)で紹介した「2つのブラウザでダイナミックHTMLを動かす その2」をネットスケープ6とMozillaで動作させる問題だ。3種類のブラウザでマーカーのように文字の位置を動かし、画面からはみ出さないように左右に往復させるスクリプトを書いてみよう。第1問と同じく、navigatorオブジェクトを使わないことが条件だ。ネットスケープ6は、オブジェクトの操作方法ではIE5以上と共通する部分が多いが、ウィンドウの幅の取得など標準化されていない部分ではナビゲーター4との互換性を残している。

ヒント ネットスケープではinnerHTML.....



[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