

HTML TIPS & TRICKS

第 25 回

誰よりも早く 最新のHTMLを使ってみよう

川田 哲 大内 勇

今月からブラウザのマークが1つ増えていることに気づいたかな? ついにインターネットエクスプローラ5.0の最新機能を使ったTIPSの登場だ。ただし、あくまでベータ版用のものなので、正式版では仕様が変わるかもしれないことに注意しよう。今月はそのほかに、IEとナビゲーターの両方で動くダイナミックHTMLや、スクリプトを高度に使いこなす方法をお届けする。



CD-ROM収録先 A Magnavi Ip9903 HtmTips
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(1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3.0以上



インターネットエクスプローラ4.0以上



インターネットエクスプローラ5.0以上



ネットスケープナビゲーター3.0以上



ネットスケープナビゲーター4.0以上



2月号「HTMLパズルに挑戦しよう」の解答

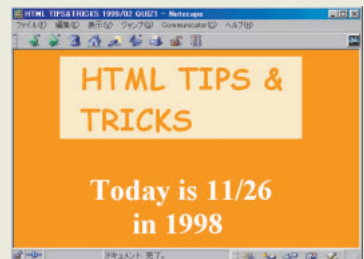
わりと新しい機能からの出題だったことを考えれば、正解者の数はまずまずと言ったところだろうか。JavaScript 1.3では、Dateオブジェクトのほかに基本的な文法やオブジェクトに変更点があるので、いつまでも古いJavaScriptを常識と考えずに新しい情報のチェックを怠らないようにしましょう。



ANSWER ① 2000年問題に対応せよ!

この問題はJavaScript 1.3を知っていれば簡単に解ける。2桁の年を得るにはgetYearを使うが、4桁の場合は「getFullYear」を使うのだ。ホームページで年月日を表示している人は、1年後に迫った2000年に対応するために書き換えておこう。

```
today = new Date();
year = today.getFullYear();
document.write(year);
```



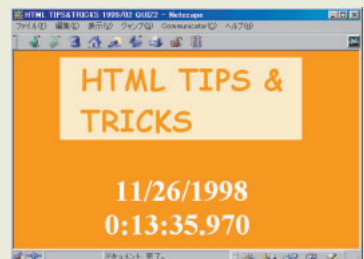
正解者: 熊本芳朗さん、良知敬介さん、田中博英さん、いのうえさん、堀江さん、近藤雄治さん、川瀬達矢さん、うおまさ@homeさん



ANSWER ② 1000分の1秒を取得せよ!

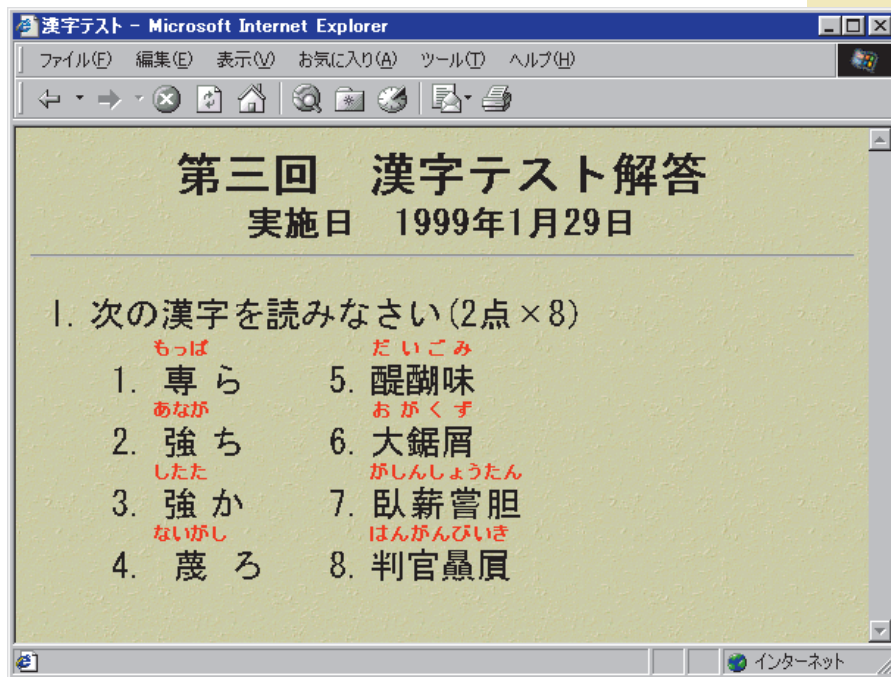
この問題も、第1問と同じくJavaScript 1.3を知っていれば簡単だ。1000分の1秒(ミリ秒)を得るには「getMilliseconds」を使う。本来3桁の数値を返すはずだが、値はハードウェアに依存するので、2桁の数値を返すマシンもあることを覚えておこう。

```
today = new Date();
msec = today.getMilliseconds();
document.write(msec);
```



正解者: 熊本芳朗さん、良知敬介さん、田中博英さん、いのうえさん、堀江さん、川瀬達矢さん

ルビをふる



ついにIE 5.0が登場した。IE 5.0 と言えば、やはりXMLへの対応がクローズアップされるが、それ以外にも新たに対応したタグやスタイルシートがある。ここではその中の1つである<RUBY>タグを紹介しよう。これは文字どおり自動的に「ルビ」をふるものだ。右のサンプルでは、この<RUBY>タグを使って漢字の上にふりがなを表示している。ルビをふることはスタイルシートを使っても不可能ではないが位置やフォントサイズなどの調節が難しい。しかし、<RUBY>タグを使えばそれも自動的に行われる。さらに、ルビ用のスタイルシートも用意されていて、ルビの配置を調節できるようになっている。



```
<RUBY>
  <RB>醍醐味
  <RT>だいごみ
</RUBY>
```

POINT

文字にルビをふるには、<RUBY>タグのほかに<RB>タグと<RT>タグも使って次のように書く。

```
<RUBY>
  <RB>ルビがふられる文字
  <RT>ルビ
</RUBY>
```

ごらんのように、<RB>タグと<RT>タグでは終了タグを省略できる。さらに<RB>タグ自体を省略することも可能だが、ルビがふられる文字であるとかかるように付けておいたほうがいいだろう。これによって、「ルビがふられる文字」の上に小さいフォントで「ルビ」が表示されるようになる。<RUBY>タグは「インライン要素」と言われるものだ。その前後は改行されないで、文章の途中にふりがな付きの漢字を入れることができる。

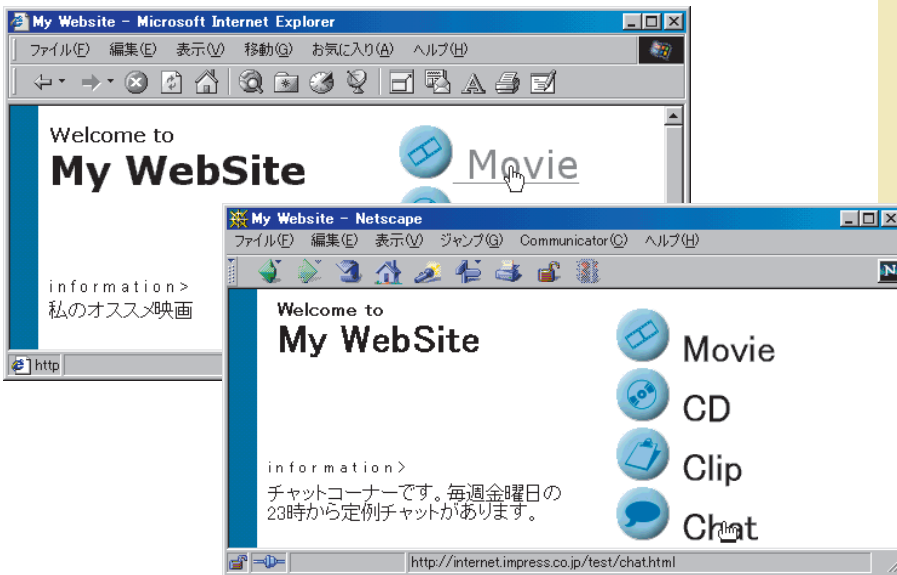
もちろん、<RUBY>タグにもスタイルシートを指定できる。<RUBY>タグに指定したスタイルは、ルビ自体にもルビのふられる文字にも反映される。また、ルビを指定する<RT>タグは、<RUBY>タグとは別にスタイルを指定できるため、もし自動的に表示したルビのサイズなどが気に入らなければ、試しに指定してみるといいだろう。右のサンプルでは、「color:red」を指定してルビを赤くしている。なお、ルビはスタイルシートでサイズを変えても自動的に位置が調節される。

<RUBY>タグにはルビ用のスタイルシートruby-alignが用意されており、ルビとルビのふられる文字の配置を変更できる。ruby-alignに指定できる値は次のとおり。

left : ルビとルビのふられる文字をともに左に寄せる。
 center : ルビとルビのふられる文字をともに中央に寄せる。
 right : ルビとルビのふられる文字をともに右に寄せる
 distribute-letter : ルビの均等割り付け
 distribute-space : ルビの均等割り付け (両端にスペースが入って均等割り付けされる)
 line-edge : ルビが短い場合は中央寄せ、ルビが長い場合は左寄せ

distribute-letterとdistribute-spaceは、ルビのふられる文字よりもルビ自体が長いときは均等割り付けされないようになっている。

2つのブラウザでダイナミックHTMLを動かす



1

```
<SCRIPT LANGUAGE="JavaScript">
var ver = navigator.appVersion.charAt(0);
var ie = 0, nn = 0;
if (navigator.appName == "Netscape") nn = 1;
if (navigator.appName ==
"Microsoft Internet Explorer") ie = 1;
function Message(s) {
if (ie && ver >= 4) {
infobox.innerHTML = s;
}
else if (nn && ver >= 4) {
document.infobox.document.open();
document.infobox.document.write(s);
document.infobox.document.close();
}
}
}</SCRIPT>
```

2

```
<DIV ID="infobox"
STYLE="position:absolute; left:32px; top:160px;">
</DIV>
```

3

```
<A onmouseover="Message('私のオススメ映画)'"
onmouseout="Message('...') HREF="movie.html">
```

POINT

このTIPSのポイントを先に述べよう。それは、「ナビゲーター4はスタイルシートでpositionプロパティを指定した<DIV>タグをレイヤーとみなす」という点だ。positionプロパティにabsoluteを指定すれば<LAYER>タグ、relativeを指定すれば<ILAYER>タグと同様なものとして扱われ、ID属性で名前を付ければレイヤーにNAME属性で名前を付けたのと同じことになる。スタイルシートで位置を指定した<DIV>タグならIE 4でも表示できるので、<LAYER>タグを<DIV>タグに置き換えれば、2つのブラウザのどちらでも座標を指定した配置ができる。

それでは上のサンプルを見てみよう。ソース①では、まずnavigatorオブジェクトを利用してブラウザの種類を判別しておく。変数「ver」にバージョン番号を入れ、変数「ie」はIEの場合は1で、そ

れ以外の場合は0、変数「nn」はナビゲーターの場合は1で、それ以外の場合は0としておく。

関数「Message」は、引数「s」に渡されたメッセージを「infobox」という名前のタグまたはレイヤーの中に表示するためのものだ。タグの位置指定は2つのブラウザで共通だとしても、その中の文字を変更するスクリプトはまったく違うものなので、先に作っておいたブラウザの種類を判別する変数をif文でチェックし、別々のスクリプトを書く。IE 4なら「infobox」のinnerHTMLプロパティに文字列を入れる。ナビゲーター4ならレイヤーのdocumentプロパティを使い、open、write、closeの各メソッドでレイヤーの内容を書き換える。

ソース②は、実際にメッセージを表示する部分の<DIV>タグだ。ID属性で「infobox」という名前を付け、STYLE属性でpositionとleft、topの各

左のサンプルは、98年9月号で紹介したマウスカーソルに反応してメッセージを表示するダイナミックHTMLを改造したものだ。読者の方からのご感想やご質問がたいへん多かったこのTIPSだが、9月号ではIE 4上でしか動作しなかった。ところが、このサンプルではナビゲーター4の上でもスクリプトエラーが出ることなく動作している。もちろん2つのブラウザのために別々のページを用意しているのではない。「互換性がないに決まっている」とダイナミックHTMLを使うのを避けている人も多いだろうが、互換性のある部分とない部分を使い分けられれば、うまく動かすためのヒントが発見できる。

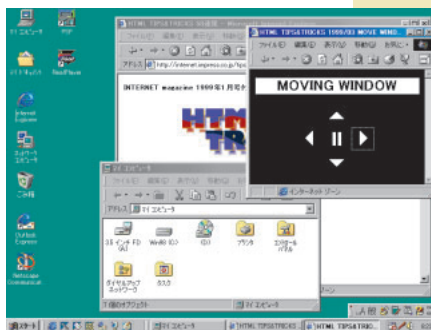
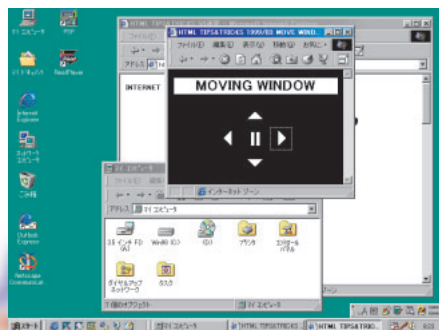
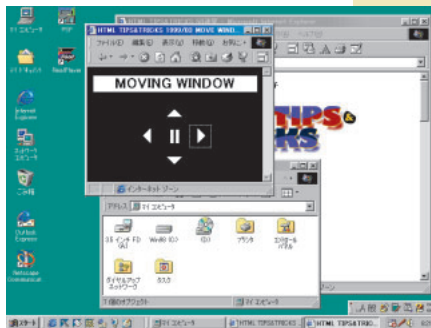
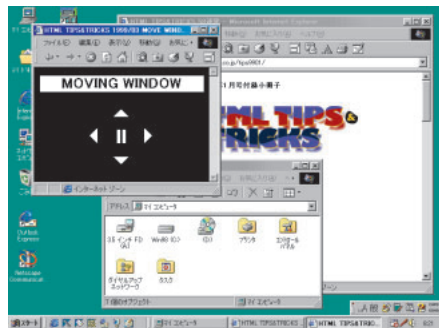
プロパティを使って好きな位置を指定する。<LAYER>タグで次のように書くのと同じになる。

```
<LAYER NAME="infobox"
LEFT=32 TOP=160>
```

ソース③は、マウスの操作で関数「Message」を呼び出してメッセージを表示させるリンクだ。onmouseoverではマウスを載せたときのメッセージを、onmouseoutではマウスが離れたときのメッセージを関数の引数に指定する。メッセージを変えて<A>タグを次々に作っていけば完成だ。

ダイナミックHTMLには互換性がないからといってあきらめるのは早い。よくよく調べれば、IEでもナビゲーターでも動くスクリプトを簡単に書く方法が見つかるかもしれないぞ。

ウィンドウを移動させる



まずは左のサンプルを見てほしい。矢印のマーク「>」をクリックすると、ブラウザのウィンドウが少しずつ右側へ移動し始める。「||」マークを押すと停止する。これはJavascriptのwindowオブジェクトを使ってウィンドウをコントロールしている様子だ。windowオブジェクトと言うと、ステータスバーに文字を表示させるTIPSがよく知られているが、このようにウィンドウ自体をダイナミックにコントロールすることもできるのだ。実際にはこのTIPSをホームページで使う機会はあまりないだろうが、遊び心で使ってみると面白いかもしれないぞ。それではさっそくこのスクリプトを見てみよう。



1

```
<A HREF="javascript:moveRight()">STOP</A>
<A HREF="javascript:stopWin()">RIGHT</A>
```

2

```
<SCRIPT LANGUAGE="JavaScript">
function moveRight() {
  window.moveBy(2,0);
  timer = setTimeout("moveRight()", 100);
}
function stopWin() {
  clearTimeout(timer);
}
</SCRIPT>
```

POINT

サンプルではウィンドウが上下左右に動くようにスクリプトを作成しているが、右方向への移動と停止の動作を行う部分のスクリプトだけを説明するとしよう。

まずはソース①のように、「RIGHT」や「STOP」という文字をクリックすることで、移動や停止のための関数を呼び出すリンクを作成する(上の画面では、Webdingsフォントを使って文字を「>」や「||」のようなマークにしている。付録CD-ROMの中のファイルを参照してほしい)。

次にソース②のように、ウィンドウの移動と停止を行う関数を作成する。関数「moveRight」では「window.moveBy」というメソッドを使っている。これはウィンドウを相対的な位置に移動させるもので、かつこ内にピクセル単位で指定した数値が

移動量になる。かつこ内は(右方向の移動量、下方向の移動量)の順序で記述する。左や上に移動させたい場合は頭にマイナスを付ければオッケーだ。

次の行には「setTimeout」がある。これは先月号でも説明したとおり、ある関数の中でsetTimeoutにその関数自身を指定すれば、繰り返し同じ処理ができる。また、同じかつこ内の数値は関数を呼び出す時間の間隔だ。関数「stopWin」でウィンドウを停止できるように、setTimeoutが返す値を「timer」という変数に入れておく。これでウィンドウを右に移動させる関数は完成だ。このサンプルでは、100ミリ秒ごとにウィンドウが右方向に2ピクセル移動することになる。

次はウィンドウを停止させる関数stopWinを見てみよう。この関数では「clearTimeout」というメソッドを使っている。これは指定した変数で表される

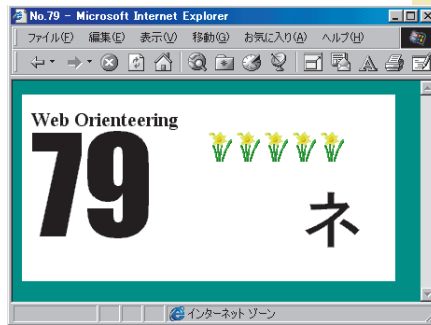
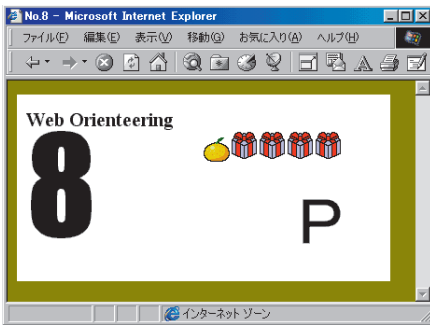
タイマー設定をクリアするもので、ここでは関数moveRightで作成した変数timerをクリアしている。これでウィンドウを停止させる関数は完成だ。さっそく動作を確認してほしい。

参考までに、ウィンドウを上下左右に移動させる処理を簡単に紹介しておこう。

```
window.moveBy(-2,0); //左に移動
window.moveBy(2,0); //右に移動
window.moveBy(0,-2); //上に移動
window.moveBy(0,2); //下に移動
```

もちろん、それぞれsetTimeoutで繰り返し処理を行うことを忘れないでほしい。

■ スクリプトレットを作る



左のサンプルは、いろいろなページを回ってキーワードやスタンプを集めていくウェブ上のオリエンテーリングをイメージしたものだ。たくさんの看板を置く必要があるが、HTMLをいちいち個別に書いていたのではめんどろだ。こうした場合でなくても、「他のページをオブジェクトとして持ってきて使えたら」と考えたことがあるかもしれない。それを可能にするのが今回ご紹介する「スクリプトレット」だ。スクリプトレットを使えば、HTMLとスクリプトをオブジェクトとして扱い、他のページから何度も利用して記述を節約できるようになる。ここでは、この便利なスクリプトレットの作り方をお伝えしよう。



1

(スクリプトレット)

```
<SCRIPT LANGUAGE="JavaScript">
var public_number;
var public_image;
var public_imagenum;
var public_key;

function public_setStamps() {
  pnum.innerHTML = public_number;
  for (i = 0; i < public_imagenum; i++)
    document.images[i].src = public_image;
  pkw.innerHTML = public_key;
}
</SCRIPT>
```

2

(スクリプトレットを使うHTML)

```
A
<BODY STYLE="background-color:#ccccff;"
  onLoad="scriptlet.number = 17;
  scriptlet.image = 'tips4b.gif';
  scriptlet.imagenum = 3;
  scriptlet.key = 'P';
  scriptlet.setStamps();">

B
<OBJECT ID="scriptlet" STYLE="width:400; height:200;"
  TYPE="text/x-scriptlet" DATA="tips4scl.htm">
</OBJECT>
```

POINT

スタイルシート(拡張子は「css」)やJavaScript(拡張子は「js」)のファイルを用意して、複数のHTMLファイルから参照するようにすれば、HTMLファイルの中にいちいちスタイルシートやJavaScriptを書く必要がなくなり、記述の節約になることは以前にも紹介してきた。IE 4以上で利用できるスクリプトレットはさらに強力だ。スクリプトレットはHTMLとスクリプトで書かれていて、一見するとページの中に別のページを表示する感じだが、スクリプトレットの変数や関数を他のページから操作できるという大きなメリットがある。これによって、同じオブジェクトでもいろいろなバリエーションを持たせることができる。

さて、そのスクリプトレットの作り方が、本誌の付録CD-ROMにスクリプトレット(Tips4scl.htm)を収録しているので見てほしい。スクリプトレット

自体は普通のHTMLファイルになっていることがわかるだろう。表示する部分は今までどおりHTMLやスクリプトで記述すればよい。そして、他のページから変更や操作をしたい変数や関数を作成する(ソース①)。変数はvarを使って名前の前に「public_」を付けて宣言する。たとえば、変数名が「val」なら「public_val」という具合だ。関数についても同じように名前の前に「public_」を付ける。今回はわざと引数を使わなかったが、もちろん関数に引数を使うこともできる。「public_」を付けた変数や関数は、スクリプトレット内ではそのまま「public_」を付けて使う。

一方、スクリプトレットを使う側のHTMLファイルでは、まず、スクリプトレットを<OBJECT>タグで指定する(ソース②のB)。ID属性で名前を付け、DATA属性でスクリプトレットのファイル名を指定

する。また、「TYPE="text/x-scriptlet"」を忘れずに付けること。

あとは、イベントが発生するときにスクリプトレットの変数や関数を指定すればよい(ソース②のA)。指定するときは、「スクリプトレットのID、スクリプトレットの変数名(関数名)」とする。ただし、スクリプトレットで宣言した「public_」付きの変数や関数は、「public_」をはずさなければならぬので注意すること。上のサンプルでは、<BODY>タグのonLoadでスクリプトレットを操作し、画像や文字を変化させている。

今回のサンプルではオブジェクトはあまり大きくないのでそれほど恩恵があるわけではないが、複雑で大きなオブジェクトをたくさん置く場合にはスクリプトレットが役に立つことになるだろう。

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

・<META>タグを制する

みなさんはHTMLのヘッダーに記述するMETAタグを活用しているだろうか？ この<META>タグは文書に関するさまざまなデータを定義するもので、シフトJISやEUCなどの文字コードを設定したり、サーチエンジンのために文書に関連するキーワードを指定する場合などに使われる。今月は<META>タグを使ってページをジャンプさせたり最新のページを表示させたりする問題に挑戦していただく。トリックがわかったらすぐに解答を送ってきてほしい。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“<META>タグを制する”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！

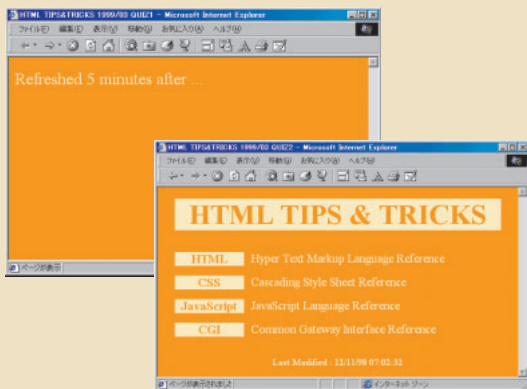
✉ ip-cdrom@impress.co.jp

なお、締め切りは2月10日とさせていただきます。



QUESTION 1

自動的にページを移動せよ！



長い間ホームページを作っていると、ページ全体の構成を変えたり、引越したりすることがある。この場合に問題になるのが訪問者に登録してもらったブックマークやお気に入りの存在だ。以前の訪問者がブックマークをたよりに再び訪れてくれても、そのページが「Not Found」ではちょっと申し訳ない気分になるのではないだろうか？ こんなときに活用できるのが今回のクイズだ。左のサンプルでは、5秒後に特定のページに自動的にジャンプさせている。解答の形式としては、「ジャンプの方法」、「秒数」、「指定ページ」のセットが正しい使い方をされていれば正解とする。みなさんからの多数の応募を期待している。

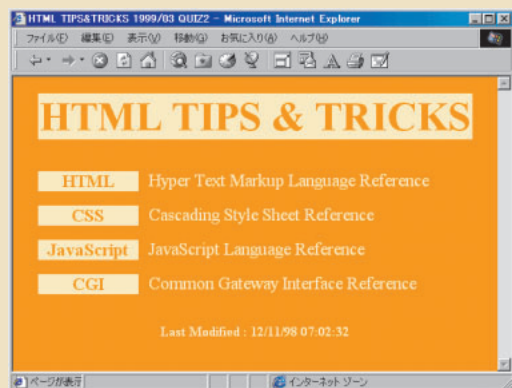


「再読み込み」、「秒数」、「URL」を指定するには……



QUESTION 2

キャッシュを無効にせよ！



WWWブラウザには、一度読み込んだページを保存しておいて、再度読み込むときにそのページを高速に表示させる「キャッシュ」という機能がある。この機能は時間を節約するのに非常に便利だ。しかし、最新のデータを表示できない場合があるという欠点もある。たとえば、画像を変更したのに古い画像が表示されてしまうといった経験のある人もいるだろう。これはキャッシュのデータが表示されていることが原因なのだ。ホームページを作っているとキャッシュされないほうが便利な場合もあるだろう。そんなときに使いたいのが今回のクイズだ。<META>タグだけで解決できる問題なので、ぜひチャレンジしてほしい！



「no-cache」と何かを組み合わせれば……



[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