

# HTML TIPS & TRICKS

第 21 回

## 誰よりも早く 最新のHTMLを使ってみよう

白畑 真 川田 哲

インターネットマガジンのリニューアルしたホームページの各コーナーでは、ありがたいことに、この連載に関するご意見が多かった。質問やご要望のある人はホームページの「Q&A」や「読者アンケート」も使ってみてほしい。「自分のページではこんなTIPSを使っている」という投稿も歓迎だ。それでは今月も1行で書けるTIPSから少々長めのスクリプトまでいろいろとお届けしよう。

インターネットマガジンホームページ URL <http://internet.impress.co.jp/>

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(9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



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



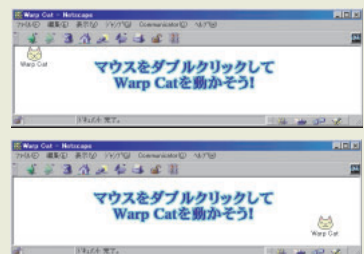
### 10月号「HTMLパズルに挑戦しよう」の解答

今回は、応募者、正答者ともに1名と少なかった。ウィンドウ全体でのダブルクリックの処理は、ナビゲーター4.0のイベントハンドラーを利用すればそれほど難しくなく、わからなかった人は、解答を見てさっそくやってみよう。いったんわかってしまえば、応用もいろいろと効くはずだ。

#### ANSWER ① ダブルクリックでレイヤーを動かせ!

ウィンドウ全体でマウスのダブルクリックを認識させるには、「document.onDbClick」にネコを動かす関数を指定する。また、ダブルクリックした位置は、関数の引数のプロパティ「ev.pageX」(X座標)、「ev.pageY」(Y座標)でわかる。

```
<SCRIPT LANGUAGE="JavaScript">
function warp(ev) {
  document.layers["neko1"].moveTo(ev.pageX, ev.pageY);
}
document.onDbClick = warp;
</SCRIPT>
:
<LAYER NAME="neko1" TOP=0 HEIGHT=0 HEIGHT=44 WIDTH=60>
<IMG SRC="image_a.gif">Warp Cat</LAYER>
```

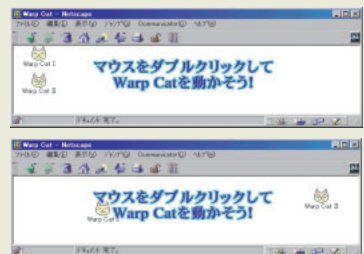


正解者: 大内勇さん

#### ANSWER ② ダブルクリックの位置を調べろ!

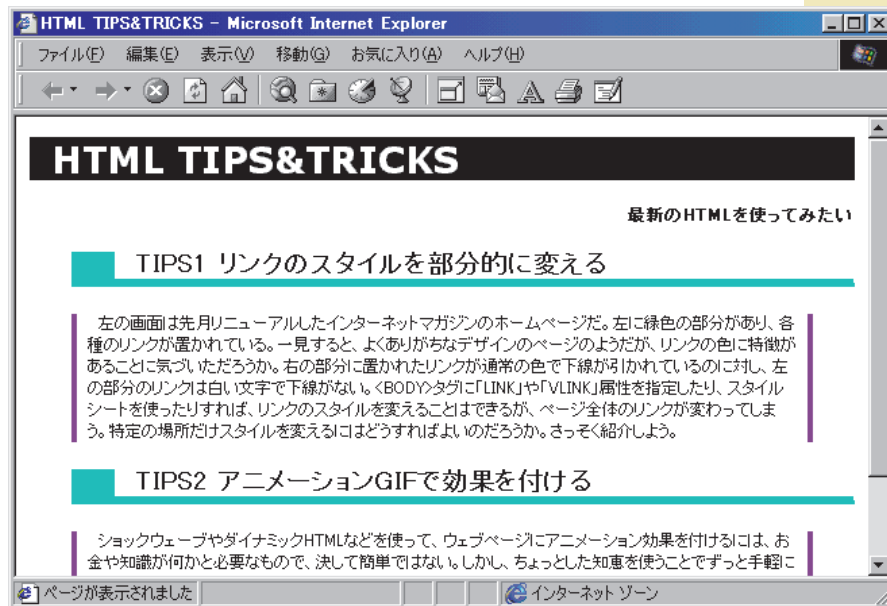
ナビゲーターでウィンドウの幅を得るには、「innerHTML」を使う。ダブルクリックした位置を「innerHTML」の半分の値と比べればよい。

```
function warp2(ev) {
  if (ev.pageX < innerWidth / 2)
    document.layers["neko1"].moveTo(ev.pageX, ev.pageY);
  else
    document.layers["neko2"].moveTo(ev.pageX, ev.pageY);
}
```



正解者: 大内勇さん

## 見出しを飾る



見出しを表すタグ<H1> ~ <H6>は、HTMLの入門書では最初のほうに出てくるが、実際に自分のページで使っている人は少ないのではないだろうか。入門書どおりに見出しを順番に付けたくても、文字が大きすぎたり小さすぎたりして、思うようなデザインにならないからだ。画像を作って見出しの代わりにしている人も多いだろう。ところが左のサンプルを見てほしい。HTMLの書き方としては、単純に<H1>タグと<H2>タグを並べているだけなのだが、1番目の見出しは黒い背景に白い文字、2番目の見出しは左と下に水色の罫線のようなものが付いている。これがわかれば、見出しのためにいちいち画像を作る必要はもうない。



```
H1 { background-color: black; color: white; border-style: none; border-width: 1;
margin-left: 0px; padding-left: 16px;
font-size: 20pt; font-weight: bold; font-family: Verdana; }
H2 { border-style: solid; border-color: #00C0C0; border-width: 0 0 6 32;
margin-left: 32px; padding-left: 16px;
font-size: 14pt; font-weight: bold; }
```

## Point

このコーナーの読者ならおわかりのように、スタイルシートを使っている。上のスタイルシートを<STYLE> ~ </STYLE>で囲い、HTMLの<HEAD> ~ </HEAD>の中に置けば、<H1>や<H2>を使った昔ながらのページが一瞬にして新しいデザインに切り替わる。

まず<H1>に指定したスタイルシートを見てみよう。背景色を黒に、文字色を白にする指定は以下の部分だ。ここは基本中の基本だ。

```
background-color: black; color: white;
```

IEでは、これでサンプルのような表示になるが、ネットスケープナビゲーターでは、文字の部分の背景が黒く塗られるだけで、サンプルのように黒い帯状にならない。そこで次のように枠線や余白の

指定を加えれば、IEと同じような表示になる。いわばナビゲーターを「ごまかす」わけだ。

```
border-style: none; border-width: 1;
margin-left: 0px;
```

あとは、「padding-left」で左に間隔を空け、「font-size」、「font-weight」、「font-family」でフォントの大きさ、太字、フォントの種類を指定する。ただし、ご存じのようにナビゲーターの場合は、日本語のページではフォントの種類は有効にならない。

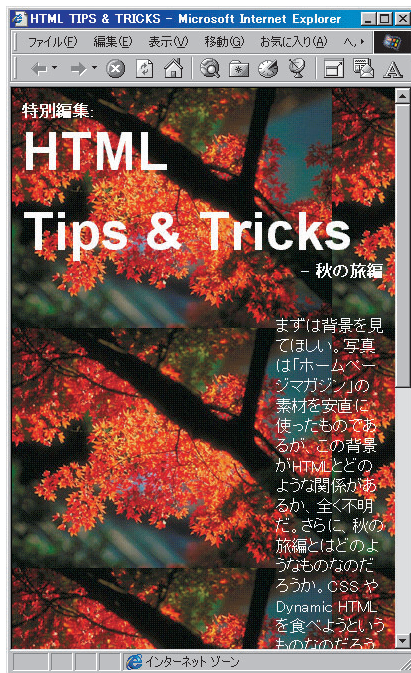
次に<H2>のスタイルシートを見てみよう。水色の飾りは、実は単なる枠線で、次のように指定している。

```
border-style: solid;
border-color: #00C0C0;
border-width: 0 0 6 32;
```

「border-style:solid」で枠線を表示させ、「border-color」で枠線の色を指定する。「border-width」には4つの数字をスペースで区切って並べ、上、右、下、左の枠線の太さを変えて指定すれば、枠線を飾りのように使うことができる。あとは、「margin-left」、「padding-left」、「font-size」、「font-weight」で余白やフォントを指定するだけだ。

スタイルシートを使えば、単純なHTMLでもこれだけのことができる。背景色や枠線を工夫して、きれいな見出しを作ってみよう。

## 背景画像を固定する



スクリプトやスタイルシートを使ったやや難しいITIPSがこれまで続いてきたが、まだ紹介していないHTMLのタグや属性もまだ残っている。ここではそうした簡単なTIPSを紹介しよう。自分のページに背景画像を指定している読者は多いと思うが、ウィンドウをスクロールすると背景画像もいっしょにスクロールされてしまうことに不満を持ったことはないだろうか。たとえば大きなJPEG画像など凝った画像を使っているページの場合、背景までスクロールされてしまい、意図しなかった空白の領域ができてしまうことがある。そこでHTMLの属性を使って背景の画像を簡単に固定する方法をお見せしよう。



```
<BODY BACKGROUND="back.jpg" BGPROPERTIES="FIXED">
```

### Point

まずは上のサンプル画面をよく見てほしい。2つの画面を比べてみると、本文の文字はスクロールされているが、背景の画像はスクロールされず、そのままの位置に固定されていることに気が付くだろう。背景は本文とともにスクロールされるものだと思っている人が多いかもかもしれないが、必ずしもそうではないのだ。

この効果を出すためには、難しいHTMLを書く必要はない、やりかたは非常に単純だ。背景の画像をBACKGROUND属性で指定している<BODY>タグに「BGPROPERTIES」属性を付けるだけで、<BODY>タグの中に「BGPROPERTIES="FIXED"」という属性を付け加えるだけで、ウィンドウをスクロールしても背景の画像はスクロールせずに固定されたままになる。

ここで重要なのは、背景にどのような画像を使うかという点だ。縦の方向にパターンが繰り返されているような画像だと、スクロールしてもまったく変化がないので、背景の画像を固定する意味がなくなってしまう。テキストチャーや小さな画像を使った場合にも、変わりはえのしない結果になってしまうだろう。そこで、ここは1つ思い切って大きなサイズの画像を背景に使ってみてはどうだろうか。大きなサイズの画像を使うと重いページになってしまうのが心配だが、圧縮率を高めにしたJPEG形式で画像を保存すれば、10数キロバイトのサイズで収まることが多い。背景にするなら画質が多少悪くてもそれほど気になることもないだろう。

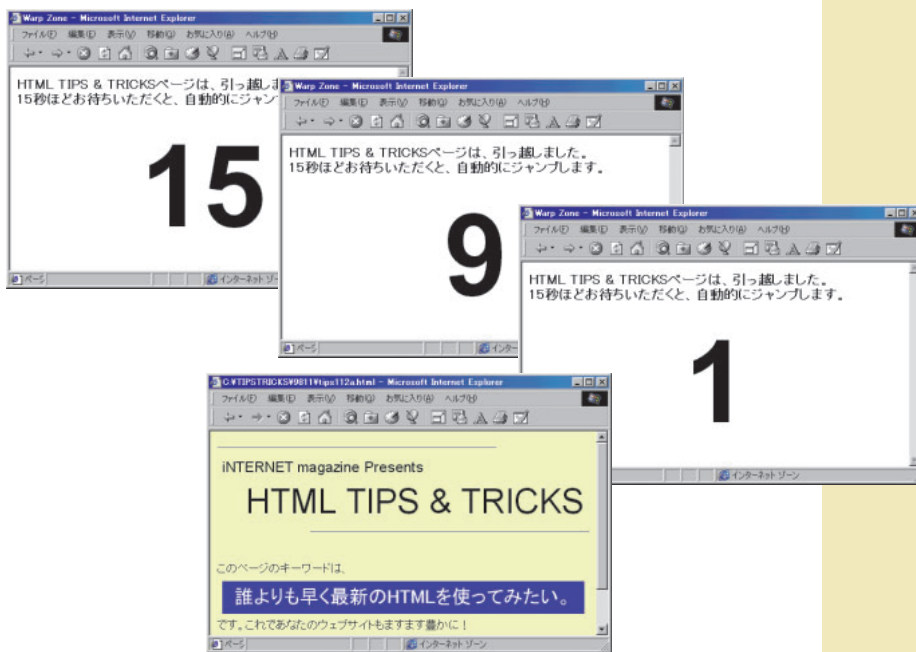
背景画像を固定する効果と1997年12月号で紹介した「画面を自動スクロールさせる」を併用す

れば、まるで映画の最後に流れるスタッフロールのような面白いページができあがる。今あるページにスタッフロールのページを付け加えてみるのも面白いかもしれない。これをヒントにユニークなページ作りにチャレンジしてほしい。

なお、背景画像を固定する効果は、以下のようにスタイルシートの「background-attachment」を使って出すこともできる。

```
<STYLE>
BODY { background-attachment: fixed;
background-image: url(back.jpg) }
</STYLE>
```

# カウントダウンタイマーを表示する



以前にこのコーナーでは、指定した日付まであと何日かをカウントダウン表示するTIPSを紹介した(1997年11月号)。今回は、1秒ごとにタイマーを表示するテクニックを紹介しよう。左のサンプルは、15秒待つと違うページへジャンプするようになっているが、ジャンプするまでに中央の数字が1秒ごとにカウントダウンして表示される。このようなタイマー表示の方法がわかれば、サンプルのような使い方のほかにもいろいろな応用がきくはずだ。もちろん15秒という数字は決まったものではなく、10秒、1分などのように任意に時間を設定して表示できるように作られている。それではさっそく見てみよう。



1

```
<SCRIPT LANGUAGE="JavaScript">
function setTime(t) {
  lastSec = t;
  tID = setInterval('graphicalTimer()', 1000);
  timerDisplay.innerHTML = lastSec;
}
function graphicalTimer() {
  lastSec --;
  timerDisplay.innerHTML = lastSec;
  if (lastSec == 0) {
    clearInterval(tID);
    location.href = "next.html";
  }
}
</SCRIPT>
```

2

```
<DIV ALIGN="CENTER" ID="timerDisplay"
STYLE="font-size:120pt; font-family:Arial;
font-weight:800;"></DIV>
```

3

```
<BODY onLoad="setTime(15);">
```

POINT

このTIPSで作成した関数は2つある(ソース①)。1つは時間を設定してタイマーを起動する関数「setTime」。もう1つは1秒ごとに数字の表示を変える関数「graphicalTimer」だ。

まずは関数「setTime」から。設定する秒数を引数の「t」から変数「lastSec」に保存する。次に「setInterval」で関数「graphicalTimer」を1000ミリ秒、つまり1秒ごとに呼び出すように指定する。「setInterval」から返ってくる「tID」は「タイマー変数」というもので、あとでタイマーをクリアするために使うことになる。設定した秒数を最初に表示しておきたいので、今までに何度か紹介してきたTIPSを使う。②のように空の<DIV>タグを用意し、IDを付ける。スクリプトで「innerHTML」に「lastSec」の値を入れて<DIV>タグの内容を変えれば、秒数が表示される。

次に関数「graphicalTimer」を見てみよう。まず変数「lastSec」の値を1つ減らし、「setTime」の場合と同じく「innerHTML」を使って表示させる。これで1秒ずつカウントダウンしていくタイマーが表示される。「lastSec」が0になったら、「clearInterval」にタイマー変数「tID」を指定してタイマーをクリアし、適当な処理を書く。ここでは違うアドレスへジャンプするスクリプトを書いている。

この2つの関数を書いたら、③のように<BODY>タグの「onLoad」で関数「setTime」を呼び出す。呼び出すときには、設定する秒数を引数に指定する。たとえば1分なら、「setTime(60)」となる。

関数「graphicalTimer」を呼び出す間隔を短く

すれば、まるでストップウォッチのような効果を出すこともできる。たとえば、関数「setTime」を次のように変える。

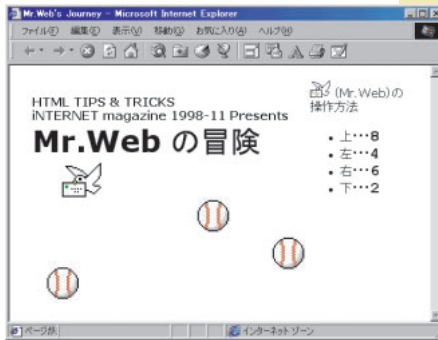
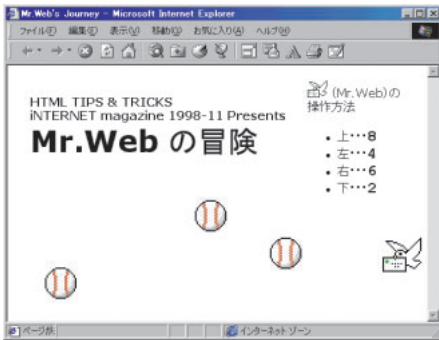
```
lastSec = t * 10;
tID = setInterval('graphicalTimer()', 100);
timerDisplay.innerHTML = lastSec / 10;
```

関数「graphicalTimer」では次のようにする。

```
timerDisplay.innerHTML = lastSec / 10;
```

こうすれば、タイマーを0.1秒ごとに表示できる。ただし、あまり短くすると、遅いマシンでは処理が追いつかなくなってしまい、タイマーが正確ではなくなるので注意しよう。

## キーボードで画像を動かす



1

```
<SCRIPT LANGUAGE="JavaScript">
function control() {
  pressKey =
String.fromCharCode(event.keyCode);
  switch (pressKey) {
    case '2':
      tori.style.pixelTop += 10;
      break;
    case '4':
      tori.style.pixelLeft -= 10;
      break;
    case '6':
      tori.style.pixelLeft += 10;
      break;
    case '8':
      tori.style.pixelTop -= 10;
      break;
  }
  document.onkeypress = control;
</SCRIPT>
```

2

```
<IMG ID="tori" SRC="tori.gif"
STYLE="position: absolute; left: 500;
top:250; width: 52; height: 46;">
```

POINT

このTIPSのポイントは、キーボード入力の処理だ。キーボード入力を処理するには、「onkey down」または「onkeypress」イベントを使う。ここでは、キーを押したときにでも画像を移動できるようにするために「onkeypress」を用いた。①のスキルの最後の行で「onkeypress」にキーボード入力を処理するための関数「control」を指定する。これにより、どれかキーを入力したときに必ず関数「control」が呼び出されるようになる。

①の中の関数「control」を見てみよう。まず「event.keyCode」から押されたキーのコードを取得する。得られたコードによって処理を分ければよいのだが、返されるコードは数値だ。キーボードのどの文字がどのコードと対応しているかを調べ、数値によって分岐のしかたを考えなければならない。しかし、そんな面倒なことをしなくても、

「String.fromCharCode」を使ってキーコードの数値を文字に変換すればよい。変換したほうが直感的にわかりやすいスクリプトを書けるはずだ。

押されたキーによって処理を分ける部分は、if ~ else で書くと煩雑になるので、switch ~ case を使ってすっきりとした形のスクリプトにした。押されたキーコードを文字に変換して変数「pressKey」に入れておき、「case」の後ろに置いた文字と「pressKey」が一致した場合に、「case」以下の処理が行われる。ここでは、「pixelTop」や「pixelLeft」を増やしたり減らしたりして画像の位置を動かしている。これは8月号の「オブジェクトの位置を自由に動かす」でも紹介した。

なお、各caseのあとで何かを処理したら、必ず「break」を入れておくことを忘れてはならない。「break」を入れないと、次のcase以降の文も処理されてしまうからだ。「break」はswitchの中か

ら抜ける命令だ。

鳥の画像は、スクリプトで移動できるようにするために、②のように<IMG>タグ内にスタイルシートを書いて「position: absolute」とし、「left」と「top」で最初の位置を指定する。ID属性で「tori」と名前を付ければ、スクリプトでは「tori.style.～」を変化させて位置を移動できる。

ここでは画像を移動しているだけだが、このスクリプトを応用すれば、キーを押したときに別の画像を出現させたり、メッセージを表示したりできる。いろいろ挑戦してみよう。



これまでこのコーナーで何度も紹介してきた画像や文字などのオブジェクトを動かすTIPSは、どれもマウスを使って動かしたり、自動的にアニメーション表示を行ったりするものだった。せっかくそこまでできるのだから、キーボードでもオブジェクトを操作したいと思うだろう。マウス操作では、かえって面倒になることもある。そこでここでは、どのキーが押されたかを判別してオブジェクトの位置を移動させるテクニックを紹介しよう。左のサンプルでは、テンキー(数字キー)を押すと鳥の画像を上下左右に動かすことができる。これを応用すれば、ホームページ上でゲームを作ることでも夢ではないはずだ。

# HTMLパズルに挑戦しよう

## 隠されたトリックを解き明かせ！



今月のテーマ

### 未対応タグを制する。

HTML4.0によってブラウザ間のタグの互換性問題が解消されようとしているが、互換性のないタグとして有名なのがネットスケープナビゲーターの<BLINK>とIEの<MARQUEE>だ。この2つはHTML4.0でもサポートされない。しかし、ダイナミックHTMLを使えば、<BLINK>と<MARQUEE>の動作を2つのブラウザで再現できる。そこで今回は、未対応タグに関するパズル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“未対応タグを制する”にチャレンジ！

### 「HTMLパズルに挑戦しよう」

宛先

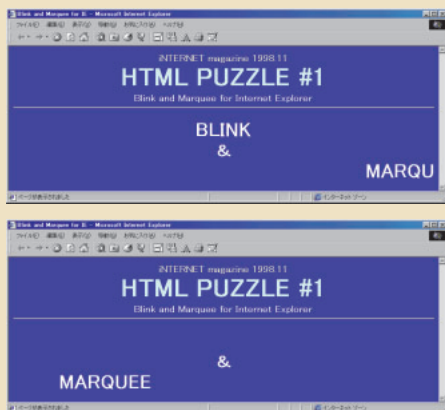
正解がわかった人も、わからなかった人も、ご意見、ご感想など何でもOK、次の宛先にメールしよう。用件の欄には必ずHTML TIPS & TRICKSの1行を忘れずに。あなたの挑戦を待つ！

✉ [ip-cdrom@impress.co.jp](mailto:ip-cdrom@impress.co.jp)

なお、締め切りは10月9日とさせていただきます。

### QUESTION 1

#### 1 エクスプローラでBLINKを実現せよ！



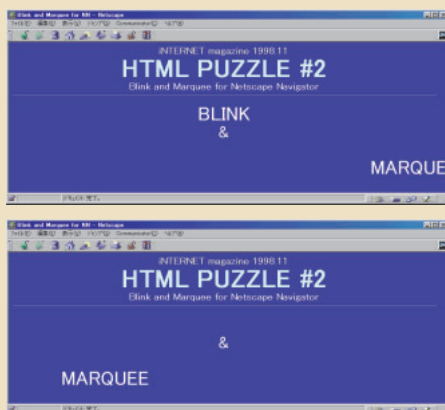
まず第1問。左のサンプルは、インターネットエクスプローラのもので、「BLINK」という文字が点滅し、「MARQUEE」という文字が右から左へと流れていく。インターネットエクスプローラで文字をスクロールさせるには<MARQUEE>タグを使えばよいが、<BLINK>タグで文字を点滅させることはできない。しかし、サンプルを見ると、対応している<MARQUEE>を使って文字を動かしているのはわかるが、あたかも<BLINK>タグを使っているかのように文字の点滅も行われている。タグにスタイルシートを指定して数行のスクリプトを書けばOKなのだが、いったいどのようにすればいいだろうか？ この問題は、比較的簡単な。



スクリプトで表示と非表示を切り替えるには……

### QUESTION 2

#### 2 ナビゲーターでMARQUEEを実現せよ！



続いて第2問。左のサンプルは第1問のものと同じだが、今度はネットスケープナビゲーターで動作している。やはり「BLINK」という文字が点滅し、「MARQUEE」という文字が右から左へと流れていく。ネットスケープナビゲーターでは<BLINK>タグを使って文字を点滅させることはできるが、<MARQUEE>タグを使って文字を動かすことはできない。スクロールを実現するには、ナビゲーター特有のレイヤーとそれを操作する数行のスクリプトを書けばよいのだが、どのようにすればいいだろうか？ マーキーの幅にはあまりこだわらなくてもよい。また、スクロールさせる文字の長さはそれほど長くなくてもよい。



スクリプトでレイヤーの位置を動かすには……



## [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](mailto:im-info@impress.co.jp)