

HTML TIPS & TRICKS

第55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

今月は遊び心いっぱいのTIPSとともに、実用的なTIPSも一緒にお送りする。手の込んだページを作ると、一般的にはユーザビリティが低下すると思われるが、最近のブラウザはユーザーの使いやすさを考えた機能が多く、それを活用することでユーザビリティを向上させられる。今月のTIPSを活用して、ぜひともそれを実現させて欲しい。



CD-ROM収録先 Magnavi Ip0109 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(6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



8月号「HTMLパズルに挑戦しよう」の解答

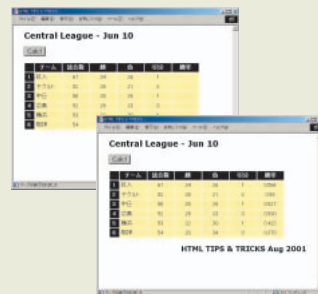
久しぶりに複雑なスクリプトを書く問題だったが、いかがだろうか。ダイナミックHTMLはアニメーションのためだけにあるのではない。この問題のように、数値をさまざまな方法で表現するためにも使えるのだ。DOMが普及すれば、データ分析のためにスクリプトを活用する場面が増えるに違いない。



ANSWER 1 勝率を計算せよ!

<TBODY>要素内の行を配列rowsで、行の中のセルを配列cellsで参照する。セル内のテキストは、firstChild.nodeValueで参照したり書き換えたりできる。

```
function calc () {
  t1 = document.getElementById("tbody1");
  for(i = 0; i < t1.rows.length; i++) {
    win = Number(t1.rows[i].cells[3].firstChild.nodeValue);
    lost = Number(t1.rows[i].cells[4].firstChild.nodeValue);
    t1.rows[i].cells[6].firstChild.nodeValue =
      String(win / (win + lost)).substr(0, 5);}
}
```



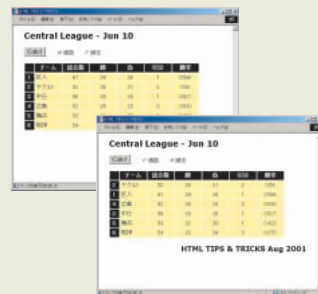
正解者: Kazuhisa Sakabeさん、ENDEさん、よしともさん、堀江さん、なるるさん、Masahiko Murataさん



ANSWER 2 表をソートせよ!

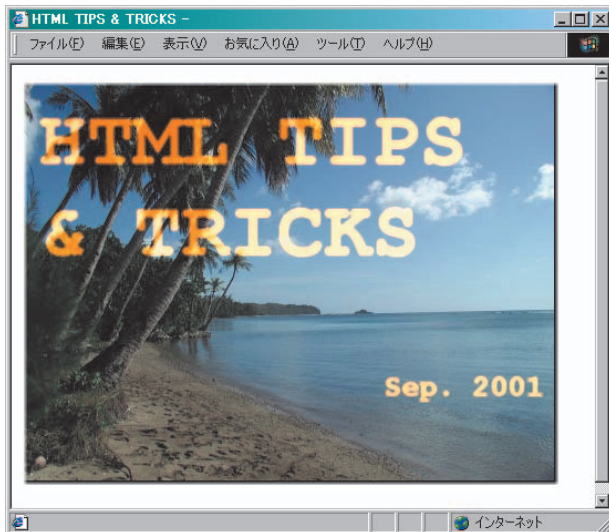
列の番号を引数にした関数を作り、ラジオボタンから「onclick="mysort(6);」のように呼び出す。行を入れ替えるには、removeChildとinsertBeforeを組み合わせる。

```
functionmysort(col) {
  t1 = document.getElementById("tbody1");
  for(i = 0; i < t1.rows.length - 1; i++) {
    pos = i;
    val = t1.rows[i].cells[col].firstChild.nodeValue;
    for(j = i + 1; j < t1.rows.length; j++) {
      val2 = t1.rows[j].cells[col].firstChild.nodeValue;
      if(val < val2) { pos = j; val = val2; }
      if(pos != i) t1.insertBefore(t1.removeChild(t1.rows[pos]), t1.rows[i]);}
}
```



正解者: Kazuhisa Sakabeさん、ENDEさん、よしともさん、堀江さん、なるるさん、Masahiko Murataさん

画像を合成する



1

```
<BODY onLoad="onLoad()">
<SCRIPT TYPE="text/javascript">
function onLoad() {
  image1.filters[0].Apply();
  image1.innerHTML = image2.innerHTML;
  image1.filters[0].Play();
}
</SCRIPT>
```

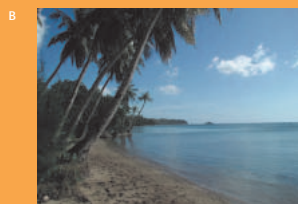
2

```
<DIV ID="image1" STYLE="filter:progid:DXImageTransform.Microsoft.Compositor(function=9);">
<DIV ID="text">HTML TIPS<BR>& TRICKS</DIV> ——— A
</DIV>
<DIV ID="image2" STYLE="display: none;">
<IMG SRC="sea.jpg" WIDTH="480" HEIGHT="360" ALT=""> ——— B
</DIV>
```

POINT

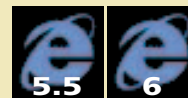
まずは、合成の対象となる2つのタグを用意する。今回はソース②のようにID属性を「text」とした<DIV>タグ(以下A)と、JPGファイルを表示させるタグ(以下B)を合成する。ID属性が「image1」の<DIV>タグでAを囲み、この<DIV>タグにCompositorフィルターを設定する。IE 5.5以降のフィルターは、「progid:DXImageTransform.....」とややこしい書き方をしたが、これはCD-ROMに収録したソースからコピーして貼り付けてほしい。一方、BはID属性が「image2」の<DIV>タグで囲み、「display: none」として非表示にしておく。

AとBを用意したら、ソース①のスク립トで合成すればいい。ID属性image1の中身は最初Aになっているが、ページを開いたときに実行される関数「onLoad」で中身をimage2のもの、つまりBに切り替える。この際に合成が行われる。



関数onLoad中のimage1.filters[0]とは、「image1の最初のフィルター」、つまりソース②で指定したCompositorフィルターだ。合成を行うには、image1の内容を変更するスク립トを「Apply」メソッドと「Play」メソッドで囲めばいい。変更前の内容と変更後の内容が合成されるのだ。今回は、ソース②で「function=9」としてAとBを「加算」(重なり合うピクセル同士の明るさを足すこと)したが、「function=」の数字を変えれば計算方法を次のように変えられる。

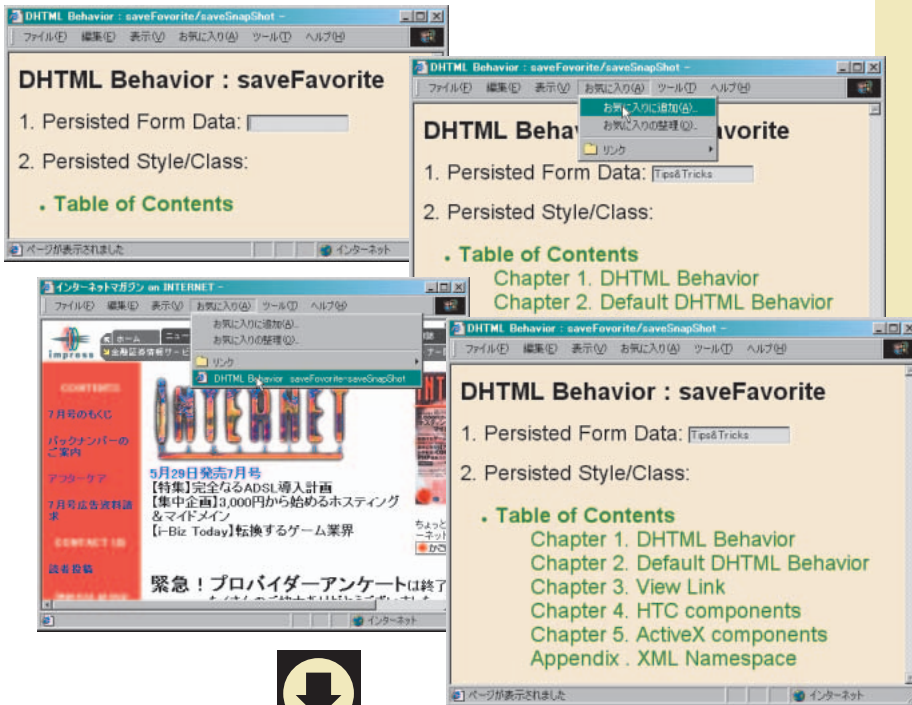
- 1 AとBのうち暗いほうを表示。
- 2 AとBのうち明るいほうを表示。
- 8 AからBを引いた値。
- 10 AとBがどちらも透明でない部分を透明にする。
- 24 AからBを引いた値。



2000年5月号の「DirectXフィルターを使う」というTIPSで紹介したとおり、IE 5.5からフィルター機能が大幅に強化されている。今回はその中でもCompositorフィルターを使って、フォトタッチソフトのように2つの画像を計算で合成する方法を紹介しよう。左のサンプルでは、「HTML TIPS & TRICKS」というテキストと画像を合成しているが、これまでの透明化フィルターなどとはひと味違った色合いになっている。もちろんテキストと画像だけでなく、画像同士やテキスト同士の合成もできるし、重ね合わせる位置をずらすことでまた違った効果も生み出せるのでいろいろと工夫してみよう。

(佐藤和人)

画面の状態を保存する



1

```
.saveFavorite {behavior:url(#default#savefavorite);}
```

2

```
function fnSaveInput(){
  oPersistInput.setAttribute("sPersistValue",oPersistInput.value);}
function fnLoadInput(){
  oPersistInput.value=oPersistInput.getAttribute("sPersistValue");}
```

3

```
<INPUT class=saveFavorite onsave="fnSaveInput()" onload="fnLoadInput()" type=text id=oPersistInput>
```

掲示板やアンケートなどのフォームに記入しているときに、どうしてもブラウザを閉じなければならなくなった。そこでお気に入りの登録して後で開いてみたが、記入した文字は消えていた。そんな経験はないだろうか。今回紹介するのは、こんな事態を防ぐTIPSだ。これを使えばIEのお気に入りへの登録、もしくは画面スナップショットの保存をすると、フォームに書かれた内容も一緒に記録できる。また、フォームのほかにもスクリプトやスタイルの状態も記憶させられるため、先月号で紹介した「エクスプローラ風のメニュー」でのメニューの状態なんていうのも保存できる。
(藤井幸孝)

POINT

このTIPSではDHTMLビヘイビアを使用している。IE5以降でビヘイビア機能はかなり強化されてきた。その1つにIEが標準で備えているDefaultビヘイビアと呼ばれる部品がある。今回はこのなかの「saveFavorite」と「saveSnapshot」の使い方を紹介しよう。DHTMLビヘイビアというと、外部スクリプトの準備が面倒で挫折した人もいるかもしれないが、デフォルトの部品を使うならそんな準備は不要だ。まずソース①のように「saveFavorite」をCSSスタイルとして登録してみよう。これは標準のビヘイビアのため、URLの記述もシンプルだ。

```
.saveFavorite
{behavior:url(#default#savefavorite);}
```

この「#savefavorite」の部分を「#savesnapshot」

やほかの名前に変えれば、そのほかのビヘイビアを使うことができる。

それではsaveFavoriteの機能を解説しよう。このビヘイビアでは、記憶したい項目(フォームのボックスやスクリプトのオブジェクト)ごとにAttributeを作成して状態を覚えさせる。記憶するタイミングにはonsave イベントを、記憶した値を呼び出すタイミングにはonload イベントを利用しよう。次にソース②を見てほしい。INPUTボックスに「class=saveFavorite」として先程作ったスタイルを適用している。これでこのタグの状態が記憶対象になる。またスクリプトで扱えるように「oPersistInput」というIDも振っておく。

onsave イベントで動作するfnSaveInput()関数はソース③のとおりだ。INPUTオブジェクトに、「sPersistValue」という名前のアトリビュートを作って、現在記入されている値(oPersistInput.

value)を記憶させる。ロード時の動作は反対にこの「sPersistValue」から値を「get」すればよい。これでフォームに文字などを入れたままお気に入りへ登録し、そこから画面を開いてやれば、フォームの文字が記入されたままになっているはずだ。実は、saveFavoriteクラスはスクリプトを書かずにクラスの設定だけでも機能する。この場合は保存するしないを細かにコントロールしたり、後で繰り返し値を取り出したりすることはできないが、簡単に作れる。くわしくはCD-ROMのサンプルを見てほしい。なお、「savesnapshot」は画面の保存時に使える。ただし<HEAD>タグの中に

```
<META NAME="save" CONTENT
="snapshot">
```

が必要になるなど、若干使い方が違う。

フォームへの入力をコントロールする



アンケートなどのフォームで、データベースなどからの出力による記入済みの項目と、新規に入力しなければならない項目とが混在してしまうことがないだろうか。このような場合は間違いがないように、記入済みの項目は入力できないようにしたいところだ。2000年11月号に紹介したdisabledを使えば入力できないようになるが、内容を送信できない。そこでreadOnlyプロパティを使った、入力できないが送信はできるというフォームを紹介しよう。また、今回のTIPSではJavaScriptとチェックボックスを利用して修正もできるようにした。ブラウザをデータベースと連動した入力フェイスとして使うときには重宝するぞ。(高橋登史朗)



1

```
function swtReadOnly(oj, idName){
  var ele = oj.form.elements[idName]
  if(ele.readOnly){ele.readOnly = false}
  else {ele.readOnly = true}}
```

2

```
<input type = "text"
  name = "NAME"
  value = "山田太郎"
  readOnly >
<input type = "checkbox"
  name = "NAMEsyuusei"
  onClick = "swtReadOnly(this, 'NAME')"
```

POINT

まずソース①を見てみよう。入力フィールドとチェックボックスだ。入力フィールドのほうはあらかじめreadOnlyが書き込まれているため、ページを読み込んだ時点で修正不可の状態に設定されている。チェックボックスはこのフィールドの修正できない状態を解除するためのものだ。つまり、はじめは修正できないが、必要ならチェックボックスをチェックして修正できるというわけだ。このチェックボックスタグの中に書かれている、

```
onClick = "swtReadOnly(this, 'NAME')"
```

の部分がこの動作を起動している部分で、チェックボックスをクリックすると、ソース①にある関数swtReadOnlyへthis(このフォームオブジェクト)と「NAME」(対象となる要素名)が渡される。それでは、この関数の中身を見てみよう。

これは、フォームの要素のリードオンリー、つまり読み出し専用で、書き換えできない状態をスイッチ切り替えるためのスクリプトだ。

```
var ele = oj.form.elements[idName]
```

これは、設定対象の要素(タグ)をeleという変数に代入している。これは別にeleという名前である必要はないが、これ以降、eleと書けばこの関数の中では、設定対象の要素を意味することになる。

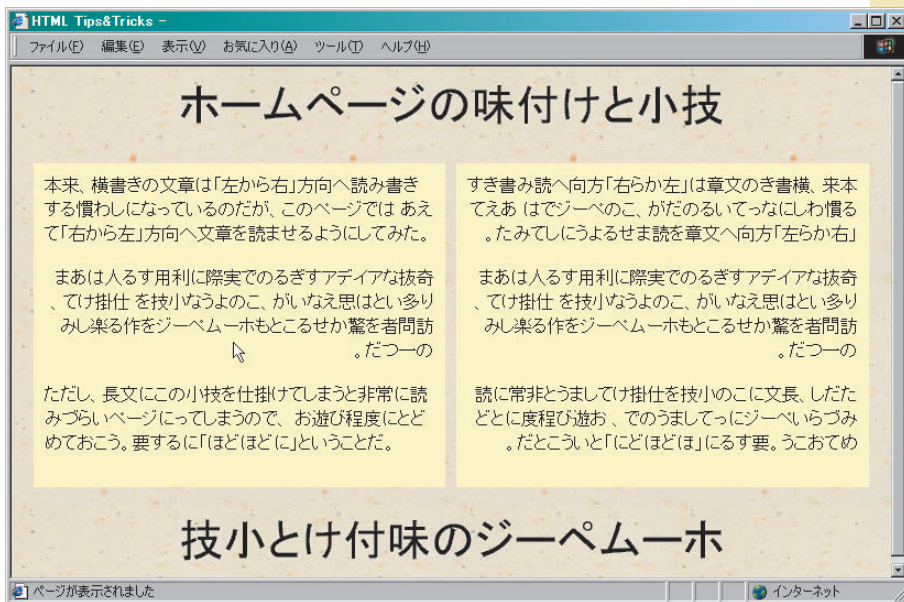
```
if (ele.readOnly){ele.readOnly = false}
else {ele.readOnly = true}
```

は、readOnlyプロパティの状態を切り替えるスイッチ部分だ。「oj.readOnly = false」で読み書きができて、trueでリードオンリーとなる。

ここでは、if(ele.readOnly)とすることで、設定対象が現在リードオンリーかどうかによって処理を分岐するようにしている。もし、クリックされたときにリードオンリーならそれを解除(false)し、そうでなければリードオンリー(true)に設定するということになる。

できれば、今回のCD-ROMサンプルを使って実際にフォームを送信し、自分の目でdisabledとの違いを確認してみよう。2つの違いを認識しないうちににげなくdisabledを使ってしまうと、せっかくフォームに内容を表示させているにも関わらず送信できなくなってしまったりして、あとで痛い目に合うかもしれない。両者の違いを認識したうえで、うまく使い分けよう。

文章の読み方向を変える



まずは左のサンプルを見てほしい。右側の文章の読み方向が左右逆転しているのがわかるだろう。これはソースを逆から書いているわけではなく、IE5から使える「読み方向を指定する」機能を使っているのだ。単純に読み方向を指定するだけならばHTMLのタグだけでもOKなのだが、今回は左側の文章にマウスポインターを載せると読み方向が左右逆転するように、JavaScriptを使ってちょっとおもしろいTIPSに仕上げてみた。奇抜なアイデアだが、このようなTIPSをホームページに仕掛けて、訪問者を驚かせることも制作者の楽しみの一つだろう。作り方はとても簡単なので、興味を持った人はぜひともチャレンジしてみよう。(大内勇)



1

```
var dirFlag = 0;
var parentObj = new Array("oDiv0");
var childObj = new Array("oBdo0");
function dirChange(n) {
  pObj = document.all(parentObj[n]);
  cObj = document.all(childObj[n]);
  if (dirFlag == 0) {
    pObj.style.direction = "rtl";
    cObj.style.direction = "rtl";
    dirFlag = 1;
  }
  else {
    pObj.style.direction = "ltr";
    cObj.style.direction = "ltr";
    dirFlag = 0;
  }
}
```

2

```
<div id="oDiv0" onmouseover="dirChange(0)" onmouseout="dirChange(0)">
<bdo id="oBdo0"> (文章) </bdo></div>
```

POINT

本来、横書きの文章は「左から右」へ読ませるようにするものだが、このページで紹介するTIPSは、あえて「右から左」へ文章をレイアウトするというテクニックだ。実用的とはいえないが、遊び心を持ってちょっと奇抜なことをやってみるとおもしろいだろう。

ソース①はマウスを載せたときに、文章の読み方向と左右のレイアウトを切り替えるJavaScriptだ。最初に2つの配列を設定し、1番目の配列にはソース②の<DIV>タグのID名を入れ、2番目の配列には同じく<BDO>タグのID名を入れる。マウスイベントで切り替える文章が複数あるときは、この2つの配列に次々と配列要素を追加していけばよい。関数dirChangeでは[dirFlag]の値によって文字のレイアウト方向と読み方向を切り替えている。[direction]プロパティが方向を表し、その値が[rtl]だと右から左への方向になり、[ltr]だと左から

右への読み方向のレイアウトになる。ソース②は読み方向とそれともなうレイアウトを適用する要素だ。<DIV>タグの内側に<BDO>タグがあることがわかるだろう。今回のTIPSの核となるのがこれだ。このタグはHTML4.0から採用されるタグで、文章の読み方向を指定する目的で用いられる。今回のサンプルではJavaScript側でレイアウト方向を指定しているため、ソース②だけでは方向を指定してはいないが、サンプルの右側のブロックは次のようにDIR属性を指定し、右から左へのレイアウトにしている。

```
<DIV DIR="rtl"><BDO DIR="rtl">
(右から左へ読ませる文章)
</BDO></DIV>
```

ソース①の説明でも触れたが、マウスイベントで

読み方向を切り替える文章が複数あるときは、ソース②全体を切り替えたい文章の数だけ追加すればよい。その場合は<DIV>タグと<BDO>タグのID名に別の名称を付け、同時にonmouseoverとonmouseoutの引数も別の値にすればよい。そしてソースの配列指定の部分に、追加した<DIV>タグと<BDO>タグのID名を追加してやればよい。例えば、[id="oDiv1"]という<DIV>タグと[id="oBdo1"]という<BDO>タグを追加した場合の配列は次のようになる。

```
var parentObj = new Array("oDiv0","oDiv1");
var childObj = new Array("oBdo0","oBdo1");
```

このテクニックを長文に対して仕掛けるとスクリーンメールが送られてくるかもしれない。ちょっとした演出程度にとどめておこう。

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

ルビを制する

IEでバージョン5から採用されたルビ(ふりがな)だが、今年の5月31日にはW3Cで仕様が正式に勧告され、XHTML 1.1のモジュールの1つとなった。Jump www.w3.org/TR/2001/REC-ruby-20010531/ これまでルビの表示はあまり知られていない機能だったが、HTMLの標準仕様となったことで、今後は古典文学などを扱ったページや子供向けのページなどで使う機会が増えてくるに違いない。そこで今月は、ルビを活用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“ルビを制する”にチャレンジ!

「HTMLパズルに挑戦しよう」宛て先

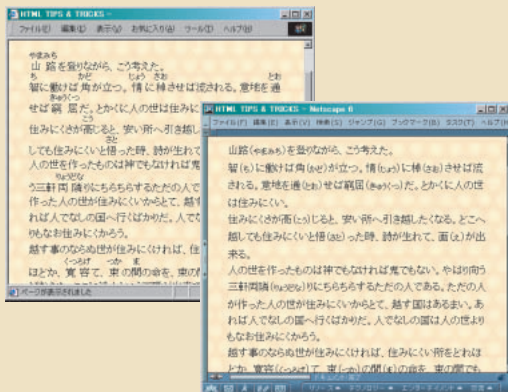
正解がわかった人も、わからなかった人も、ご意見、ご感想など何でもOK、次の宛て先にメールしよう。用件の欄には必ず
HTML TIPS & TRICKS
の1行を忘れずに。あなたの挑戦を待つ!

✉ im-html@impress.co.jp

なお、締め切りは8月10日とさせていただきます。

QUESTION 1

かっこを使いこなせ!

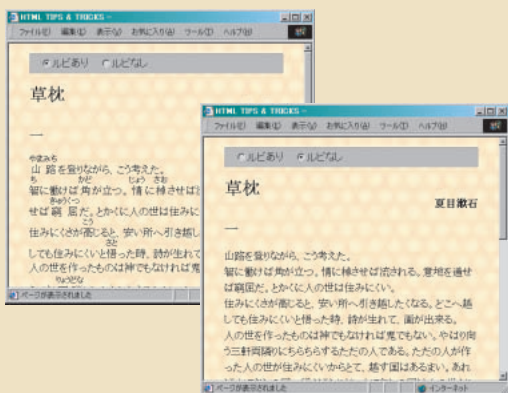


ルビのような新しいHTMLの機能を使うときには、ブラウザの互換性が気になるもの。ルビに対応していないブラウザでページを開くと、「漢字かんじ」のように表示されてしまって格好が悪い。この問題を回避するために、ルビの仕様にはある仕掛けが組み込まれている。左のサンプルではその仕掛けを使って、IEでは「漢字かんじ」のようにルビを正しく表示させ、ネットスケープでは「漢字(かんじ)」のようにかっこの中に入れて表示させている。つまり、IEで自動的にルビのかっこを表示させないようにできるのだ。これならどのブラウザでも自然な表示になる。W3Cのページでルビの仕様書を見てその方法を探してみよう。

ヒント ("と") をあるタグで囲う.....

QUESTION 2

ルビの有無を切り替える!



ウェブページでルビを表示できるのは確かに便利だが、漢字の読みをよく知っている人にとってはかえって読みにくいページになってしまうかもしれない。ルビもタグの一種なのだから、スタイルシートやスクリプトを適用できるはずだ。そこで第2問では、左のサンプルのように、ページ上部の「ルビあり」と「ルビなし」をチェックすることで、ルビを表示させたり消したりするスクリプトを書いてみよう。スクリプトは短ければ短いほど解答として好ましい。ページ中のルビをすべて列挙して、1つ1つ削除したり表示を消したりするようなことはしないでほしい。スタイルシートを活用して一発で切り替えるスクリプトを考えよう。

ヒント スタイルシートの有効と無効を切り替える.....



[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