

森田 雄の

正しいXHTML塾

Movable Typeに間違っただけのページを作らせないための公式

主にプログツールなどの導入により、普及が進んでいるXHTMLですが「本当に正しい」XHTMLの書き方を、あなたは理解していますか？ 全2回で解説する「正しいXHTMLの書き方」後編となる今回は、少し高度な正しいXHTMLの書き方の公式と、実際にエントリーを書く場合の例を紹介します。前号に引き続き、正しいXHTMLを実現した、オリジナルのMovable Type完全テンプレートセットをベースとしていますので、ダウンロードして使いながら試してください。

講師 森田 雄

1975年生まれ。東芝EMIでマルチメディアCD-ROMのデザインおよびオーサリングに多数携わった後、独立してフリーランスとなる。1998年よりマイクロソフトに在籍し、モバイル戦略に携わる。2000年、XMLをはじめとする多様なプロトコルの専門家としてビジネス・アーキテクツの設立に参画。現在、ユーザビリティおよびアクセシビリティのスペシャリストとして活躍中。 yumor@b-architects.com
URL <http://www.b-architects.com/>

上級編XHTMLマークアップの「公式」

前号では、正しいXHTMLを書くことがなぜ大切なのかの理由と、基本的なページ要素に関して正しいXHTMLを書くための「公式」を解説しました。前回解説した

公式は、文章の基本マークアップ、強調、引用、箇条書き、リンク、アスキーアート、見出しなどがあり、さらにXHTMLの基礎的な知識も解説しました。

後編では、少し高度な公式を説明しますので、XHTMLの基本を理解していない場合は、まず前編の内容を確認してから後編に入ってください。

【公式11】: 変更(挿入・削除) 要素・ins要素

公開した記事にあとから文章を挿入・削除する場合、挿入した部分をins要素で、削除したと示したい部分をdel要素でマークアップすれば、どのように変更されたのかを明らかにできます。挿入・削除は、ブロックレベル要素ごとでもできます。

ins要素とdel要素には、挿入・削除した日時を記すためのdatetime属性というものが、「yyyy-mm-ddThh:mm:ss+09:00」のように書きます(日本標準時の場合)。「T」はそのまま「T」と書き、それ以外には年月日と時分秒を入れます。datetime属性はなくても構いませんが、入れておいたほうが情報の付加価値は増します。

```
<p>夕食の<del>ビーフカレー</del><ins>マトンカレー</ins>は、肉がすごくおいしかった。</p>
<p>前日から<ins datetime="2004-01-28T05:38:00+09:00">ニンニクに漬け込んで</ins>下ごしらえをしていただけのことはある。<del datetime="2004-01-28T05:38:00+09:00">少し固かったが。</del></p>
<ins><p>おいしいカレーなら何日でも続けて食べられる。</p></ins>
```

▼ このXHTMLをブラウザで表示すると……

夕食の~~ビーフカレー~~マトンカレーは、肉がすごくおいしかった。

前日からニンニクに漬け込んで下ごしらえをしていただけのことはある。~~少し固かったが~~。
おいしいカレーなら何日でも続けて食べられる。

【公式12】:ルビ(ふりがな)< ruby 要素 >

XHTML 1.1にはルビモジュールが導入されているので、ルビ(ふりがな)を使えます(右図上)。ルビ表示に対応していないブラウザでは、ルビ対象になる文字列に続いてルビをカッコで囲んで表示するようにも指定できます(右図下)。

ルビを付けたい漢字などを<rb> ~ </rb>で囲み、ルビが表示できないときに使いたいカッコの開き部分を<rp> ~ </rp>で、ルビを<rt> ~ </rt>で、閉じカッコをこちらも<rp> ~ </rp>で指定して、その全体を<ruby> ~ </ruby>で囲みます。

```
<p>こちらこそ<ruby><rb>夜露死苦</rb><rp></rp><rt>ヨロシク</rt><rp></rp></ruby></ruby>お願いします。</p>
```

▼このXHTMLをブラウザで表示すると……

ヨロシク
こちらこそ夜露死苦お願いします。

こちらこそ夜露死苦(ヨロシク)お願いします。

XHTML 1.1準拠のフルセット

Movable Type用オリジナルテンプレート

XHTML 1.1に準拠したオリジナルのMTテンプレートを、本稿の付録として用意しています。このテンプレートを使えば、ページのXHTMLは「正しいXHTML」になります。あとは、記事の本文に正しいXHTMLを書けば完成です。

トップページ用のMain Indexとスタイルシートだけでなく、各アーカイブページやコメントページ用のテンプレートが含まれたフルセットになっています。

なお、Stylesheetテンプレートでの表示は、ウィンドウズ版のIE 6、Mac版のIE 5.1以降、Safari 1.0以降、Mozilla 1.5以降で見ることを想定して構成されています。



注意

このオリジナルテンプレートはカテゴリー設定を処理するように作られているため、正しく使用するためにはMovable Typeのブログの設定でカテゴリーアーカイブが有効になっている必要があります。標準の状態ではカテゴリーアーカイブは有効になっていないので、その状態でブログを再構築するとエラーが発生してしまいます。

カテゴリーアーカイブを有効にするには、Movable Typeの管理画面でオリジナルテンプレートを使用するブログの管理画面を開き、[ブログの設定]>[アーカイブの設定]>[アーカイブの種類]にある[Category]にチェックを入れます。

ダウンロードできるテンプレート

インデックス・テンプレート

- Main Index
- Stylesheet

アーカイブに関連したテンプレート

- Category Archive
- Date-Based Archive
- Individual Entry Archive

その他のテンプレート

- Comment Preview Template
- Comment Error Template
- TrackBack Listing Template

テンプレートのダウンロードは
インターネットマガジンの
ウェブサイトから！

URL <http://internet.impress.co.jp/im/200410mt/>

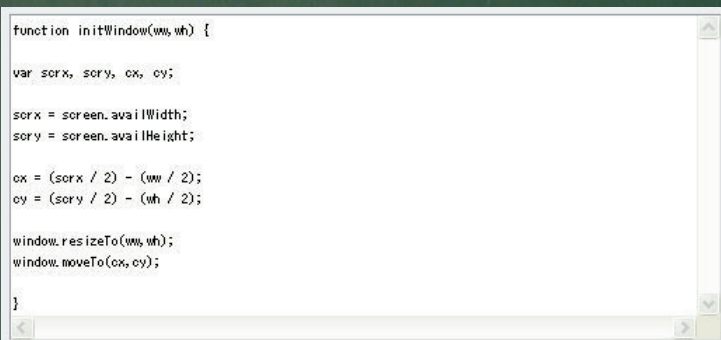
【公式13】: プログラムコード <pre 要素 + code 要素>

HTMLでは連続した半角スペースは意味を持ちませんが、プログラムコードではインデントなどの用途で使われるために、意味を失われては困ることがあります。そこで、書いたとおりの改行や空白がそのまま表示に反映される整形済みテキストのpre要素とし、さらにコードを表すcode要素を使います。

付録のStylesheetテンプレートでは、バグ対策のためにMac版IE 5では他のブラウザとは違うスタイルが設定されています。内容がボックス幅を超えないように、各行に適宜改行を入れるようにしましょう。また、整形済みテキストにプロポーショナルフォントを指定していますが、プログラムコードには等幅フォントを指定しています。

```
<pre><code>function initWindow(ww, wh) {  
  var scrx, scry, cx, cy;  
  scrx = screen.availWidth;  
  scry = screen.availHeight;  
  cx = (scrx / 2) - (ww / 2);  
  cy = (scry / 2) - (wh / 2);  
  window.resizeTo(ww, wh);  
  window.moveTo(cx, cy);  
}</code></pre>
```

▼ このXHTMLをブラウザで表示すると.....



```
function initWindow(ww, wh) {  
  var scrx, scry, cx, cy;  
  scrx = screen.availWidth;  
  scry = screen.availHeight;  
  cx = (scrx / 2) - (ww / 2);  
  cy = (scry / 2) - (wh / 2);  
  window.resizeTo(ww, wh);  
  window.moveTo(cx, cy);  
}
```

【公式14】: 画像の埋め込み

画像を配置する場合は、p要素の中にimg要素を入れます。画像には縁取りが出ないスタイルになっているので、縁取りが欲しいときは、画像自体を加工するようにしてください。

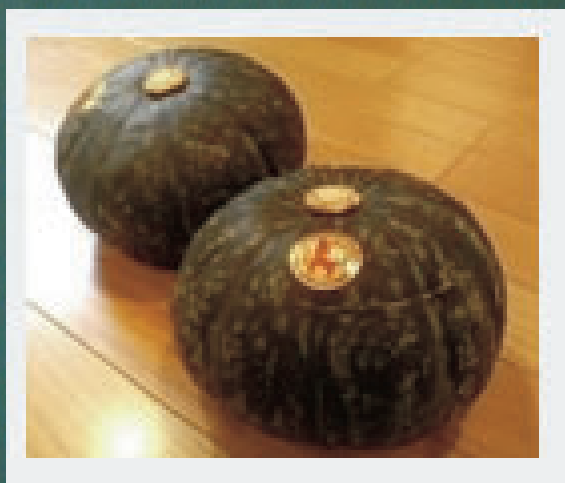
src属性は画像ファイルの場所をルートディレクトリーからの絶対パスで書くといいでしょう。右の例は、サーバー上にある「images」ディレクトリーに画像ファイルが置いてある場合の書き方です。

width属性には画像の幅を、height属性には画像の高さをそれぞれピクセル数値で書き入れます。ウィンドウズ版IEのバグ対策のために、style属性中の2つの数値にも、それぞれwidth属性とheight属性に書き入れたピクセル数値を書いてください。

alt属性には、src属性で指定された画像によって得られる情報を「代替テキスト」として文章で書き入れます。

```
<p></p>
```

▼ このXHTMLをブラウザで表示すると.....



複数の画像を横に並べたい場合は、同一のp要素内で、img要素を複数書きます。縦に並べたい場合は、br要素で改行するのではなく、1つずつp要素で囲みましょう。これに限らず、基本的にはbr要素を使わないようにマークアップすると、XHTMLとして美しくなります。



「<」や「>」「&」「<」「>」「"」は文字実体参照で

「<」や「>」の記号は、XHTMLのタグに使う特別な記号なので、文章中にそのまま書くとページがおかしくなります。MTの自動整形を使うと自動的に処理してくれますが、自動整形を使わずにエントリーを書く場合には、これらの記号を書くときは「文字実体参照」という仕組みを利用して書き換える必要があります。

具体的には「<」は「<」、>」は「>」、"」に書き換えると、ブラウザで表示される際にはそれぞれ「<」「>」と表示されるようになります。アンド記号（&）もまた特別な記号なので、通常の文章などに「&」を書きたいときは「&」

と書きます。二重引用符号（"）は、通常はそのまま"と書いていいのですが、画像のalt属性などの属性値の中に"を書くときは、「"」と書く必要があります。属性値は"で括って記述するため、中に"があると問題が生じるのです。

【公式15】: 画像に回り込むテキスト <p要素とimg要素の属性指定>

画像をただ配置するのではなく、続くテキストを回り込み表示させる場合の公式です。画像の左側にテキストが回り込むものはimgRightクラス、右側にテキストが回り込むものはimgLeftクラスとして、2とおりのスタイルを用意しています。

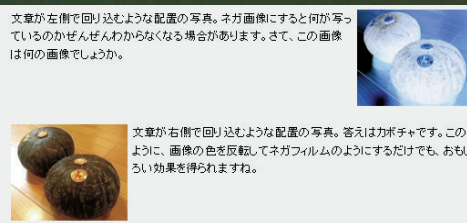
この「公式」は、ウィンドウズ版IEなどいくつかのブラウザのバグ対策を含んだものになっています。img要素のheight属性に書いたのと同じ値を、p要素のstyle属性内の該当する3か所にも書きます。さらに、img要素ではwidth属性とheight属性に加えて、style属性にwidthプロパティとheightプロパティを書きます。

回り込みテキストはp要素以外では設定できません。

```
<p style="min-height:105px;height:105px;/**/height:auto;/**/_height:105px;" class="withImage">文章が左側で回り込むような配置の写真。ネガ画像にすると何が写っているのかぜんぜんわからなくなる場合があります。さて、この画像は何の画像でしょうか。</p>
```

```
<p style="min-height:105px;height:105px;/**/height:auto;/**/_height:105px;" class="withImage">文章が右側で回り込むような配置の写真。答えはカボチャです。このように、画像の色を反転してネガフィルムのようにするだけでも、おもしろい効果を得られますね。</p>
```

▼ このXHTMLをブラウザで表示すると……



回り込みテキストのスタイルはなぜ複雑なの？

画像に回り込むテキストのp要素のstyle属性が複雑なのは、ブラウザごとに違うスタイルシートの解釈を吸収するためです。

```
style=" min-height:105px; height:105px; /**/height:auto;/**/ _height:105px;"
```

ウィンドウズ版IE 6 (WinIE6)やMac版IE 5 (MacIE5)などは、ボックスの最小の高さを指定するmin-heightプロパティを解釈しません。通常、min-heightに絶対値の長さを指定するときは、heightプロパティに「auto」を指定しておけば、ボックスサイズがmin-heightよりも大きくなる場合は、ボックスサイズは自動

的に大きくなります。ところが、WinIE6やMacIE5は、min-heightプロパティを解釈せず、heightプロパティがmin-heightプロパティであるかのように動作します。heightに絶対値の長さを指定していると、内容が高さを超える場合、ボックスサイズが勝手に広がってしまうのです(仕様と反する動作です)。そこで、同じプロパティを複数回指定するとあとから読み込んだ値を使うというCSSのルールと、ブラウザがCSSを読み込む際のバグを利用して、ブラウザごとに違う値を指定しているのがこの公式です。p要素のstyle属性には、まずmin-heightで内

容となる画像の縦幅を指定します()。次にmin-heightを解釈しないブラウザ用に、heightで同じ縦幅を指定します()。min-heightを正しく理解するMozillaなどのブラウザでは、heightに指定された絶対値の長さ以上にはボックスサイズが広がらないので、「auto」を指定してheightプロパティを上書きします。ただし、ここで、MacIE5のコメント記述の解釈バグを利用して、この記述をMacIE5にだけ読めないようにします()。この指定でWinIE6はheight:autoを理解するので、最後にWinIE6が解釈できる書き方でheightの指定を上書きしています()。

実際にエントリーをマークアップしてみよう

マークアップ例1： 簡単なエントリー

さて、「公式」はだいたい頭に入ったでしょうか。実際のエントリーのマークアップは、エントリーに書こうと思っている日本語の文章をまず書き、その文章を、「公式」を利用してマークアップしていくような流れで実現できます。

たとえば、 のような日記のエントリー用の文章があったとします。

この文章には、小さな段落が2つあります。「名古屋」から「なりました。」までが1つの段落、残りがもう1つの段落です。これらは通常の文章なので、どちらもp要素で意味づけします()。

そして、実際にぱんだカーに乗っている写真が手元にありますので、せっかくですからそれを挿絵にしたいとします。1つ目の段落のところが、ちょうどその写真に見合う内容なので、1つ目のp要素を、画像への回り込みテキストで利用するp要素に変えてみます()。

このマークアップ済みのものに、 のようにタイトルを付けたりカテゴリを付けたりしてエントリーとして保存すれば、 のようになります。

p要素で段落をマークアップ

```
<p>名古屋・松坂屋本店の屋上で、古びたぱんだカーを発見しまして、すわ今こそとばかりに乗ってみました。なかなかの乗り心地で、良い思い出になりました。ちなみに松坂屋へはコレに乗るために行ったのではなくて、もちろん昼ごはんとかを適当に食べるために行ったわけなのですが、ちょっとレストラン街のある階の構造が微妙な感じで、やや道に迷い気味で屋上へ出てしまったのでした。結果オーライ。
```

p要素で段落をマークアップ

```
<p>名古屋・松坂屋本店の屋上で、古びたぱんだカーを発見しまして、すわ今こそとばかりに乗ってみました。なかなかの乗り心地で、良い思い出になりました。</p>
```

```
<p>ちなみに松坂屋へはコレに乗るために行ったのではなくて、もちろん昼ごはんとかを適当に食べるために行ったわけなのですが、ちょっとレストラン街のある階の構造が微妙な感じで、やや道に迷い気味で屋上へ出てしまったのでした。結果オーライ。</p>
```

画像の埋め込みと回り込みをマークアップ

```
<p style="min-height:105px;height:105px;/***/height:auto;/**/_height:105px;" class="withImage">名古屋・松坂屋本店の屋上で、古びたぱんだカーを発見しまして、すわ今こそとばかりに乗ってみました。なかなかの乗り心地で、良い思い出になりました。</p>
```

```
<p>ちなみに松坂屋へはコレに乗るために行ったのではなくて、もちろん昼ごはんとかを適当に食べるために行ったわけなのですが、ちょっとレストラン街のある階の構造が微妙な感じで、やや道に迷い気味で屋上へ出てしまったのでした。結果オーライ。</p>
```

タイトルやカテゴリを指定

タイトル	主カテゴリ
<input type="text" value="ぱんだカー"/>	<input type="text" value="LIFE"/>
<small>複数のカテゴリを設定する</small>	
エントリーの内容(body)	
<pre><p style="min-height:105px;height:105px;/***/height:auto;/**/_height:105px;" class="withImage">名古屋・松坂屋本店の屋上で、古びたぱんだカーを発見しまして、すわ今こそとばかりに乗ってみました。なかなかの乗り心地で、良い思い出になりました。</p><p>ちなみに松坂屋へはコレに乗るために行ったのではなくて、もちろん昼ごはんとかを適当に食べるために行ったわけなのですが、ちょっとレストラン街のある階の構造が微妙な感じで、やや道に迷い気味で屋上へ出てしまったのでした。結果オーライ。</p></pre>	

正しいXHTMLのエントリーが完成

ぱんだカー

2004年06月25日 16:16

(LIFE)



名古屋・松坂屋本店の屋上で、古びたぱんだカーを発見し、すわ今こそとばかりに乗ってみました。なかなかの乗り心地で、良い思い出になりました。

ちなみに松坂屋へはコレに乗るために行ったのではなくて、もちろん昼ごはんとかを適当に食べるために行ったわけなのですが、ちょっとレストラン街のある階の構造が微妙な感じで、やや道に迷い気味で屋上へ出てしまったのでした。結果オーライ。

Posted by yuu

[Comments \(1\)](#) [TrackBack \(2\)](#)



マークアップ例2： 複雑な構造のエントリー

もう一つ、より多くの要素でマークアップする文章の例を見てみましょう。

最初の行は通常の文章(p要素) 続く「以下のコード」の3行はプログラムコード、その次にまた通常の文章(p要素) ときます。HTML4.01のDTDを引用している部分は、プログラムコードをブロックレベルとして引用しましょう。そして最後に通常の文章(p要素)のようにマークアップできます。プログラムコード中に特別な記号である「<」「>」が含まれていますので、これらはそれぞれ文字実体参照に書き直しておく必要があります。

日本語の文章さえあれば、文章を構成する各要素を適切なXHTMLの要素で意味づけすることで、正しいXHTMLで書かれたエントリーにできます。多少時間はかかりますが、それだけ文章を推敲する時間も得られて良い文章になるでしょう。正しいXHTMLになることに加えて、ウェブログの内容も充実させられますね。

今回の公式で物足りなくなったら、自分で「正しいXHTMLの公式」を作ってみましょう。その際に役に立つ情報源を下に紹介しますので、ぜひ参考にしてください。

マークアップ前の文章

```
以下のコードは、HTML 4.01 Strictとして妥当です。
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<title>妥当な文書のタイトル</title>
<p>妥当な文書のパラグラフ</p>
html要素の開始タグと終了タグは書かれてませんが、それは省略可だからです。そしてhtml要素の内容は、
head要素、body要素の順で1つずつ必須であるとDTDに書かれていますが、やはりどちらの要素の開始タグも
終了タグも書かれていません。これもまたそれぞれ省略可だからです。
<!ENTITY % html.content "HEAD, BODY">
<!ELEMENT HTML 0 0 (%html.content;) -- document root element -->
<!ATTLIST HTML
  %i18n; -- lang, dir --
  >
「HTML 4 Document Type Definition」から
しかしhtml要素もhead要素もbody要素も、いずれも要素は存在しています。
DOCTYPE宣言の直後からtitle要素の終了タグまでがhead要素であり、p要素の開始タグから終了タグまで
の範囲がbody要素なわけです。要素が必須か否かと、タグが省略できるか否かは、同列のコトじゃないのです。
```

段落、コード、引用、文字参照をマークアップ

```
<p>以下のコードは、HTML 4.01 Strictとして妥当です。</p>
<pre><code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"&gt;
&lt;title&gt;妥当な文書のタイトル&lt;/title&gt;
&lt;p&gt;妥当な文書のパラグラフ&lt;/p&gt;</code></pre>
<p>html要素の開始タグと終了タグは書かれてませんが、それは省略可だからです。そしてhtml要素の内容は、
head要素、body要素の順で1つずつ必須であるとDTDに書かれていますが、やはりどちらの要素の開始タグも
終了タグも書かれていません。これもまたそれぞれ省略可だからです。</p>
<blockquote cite="http://www.w3.org/TR/html401/sgml/dtd.html#html.content">
<div class="quote">
<pre><code>&lt;!ENTITY % html.content "HEAD, BODY"&gt;
&lt;!ELEMENT HTML 0 0 (%html.content;) -- document root element --&gt;
&lt;!ATTLIST HTML
  %i18n; -- lang, dir --
  &gt;</code></pre>
</div>
<ul class="cite">
<li><code>&lt;a href="http://www.w3.org/TR/html401/sgml/dtd.html#html.content">
HTML 4 Document Type Definition</a> から</code></li>
</ul>
</blockquote>
<p>しかしhtml要素もhead要素もbody要素も、いずれも要素は存在しています。</p>
<p>DOCTYPE宣言の直後からtitle要素の終了タグまでがhead要素であり、p要素の開始タグから終了タグまで
の範囲がbody要素なわけです。要素が必須か否かと、タグが省略できるか否かは、同列のコトじゃないのです。</p>
```

正しいXHTMLをさらに知るために

HTML 4.01仕様書

W3CのHTML 4.01仕様書の邦訳です。XHTMLの元になった文書構造や要素・属性の考え方、使い方などが詳細に解説されています。

<http://www.asahi-net.or.jp/~sd5a-ucd/rec-html401j/cover.html>

XHTML 1.0: 拡張可能ハイパーテキストマークアップ言語

W3CのXHTML 1.0仕様書の邦訳です。これを読んでHTML 4.01との違いを理解しましょう。

<http://www.doranekeo.org/webauth/xhtml1/02000126/Overview.html>

XHTML 1.1 - モジュールベースXHTML

W3CのXHTML 1.1仕様書の邦訳です。XHTML 1.0とXHTML 1.1の違いが主な内容です。

<http://www.doranekeo.org/webauth/xhtml1/20010531/Overview.html>

Cascading Style Sheets, Level 2

W3CのCSS 2.0仕様書の邦訳です。正しいXHTMLをCSSでデザインしてみましょう。正しいXHTMLである必要性も、ここから学べます。

http://www.y-adagio.com/public/standards/tr_css2/toc.html

ユニバーサルHTML/XHTML

毎日コミュニケーションズ刊 / 360p / 2,310円 / ISBN4-8399-0454-5

詳解 HTML&XHTML&CSS辞典
秀和システム刊 / 566p / 2,100円 / ISBN4-7980-0221-6

スタイルシート スタンダード・デザインガイド
毎日コミュニケーションズ刊 / 303p / 2,730円 / ISBN4-8399-1501-6

スタイルシート・スタイルブック
翔泳社 / 207p / 2,310円 / ISBN4-7981-0585-6



[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