

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により導入が進んでいる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」前編となる今回は、XHTMLの基礎、正しいXHTMLのためのMovable Typeの設定、正しいXHTMLの書き方の基本的な公式を紹介します。正しいXHTMLを実現した、オリジナルのMovable Typeの完全テンプレートセットも紹介します。

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/>

XHTMLの「正しさ」って何？

XHTMLは難しくない！

XHTMLは、W3C(インターネットで利用される技術の標準化を進める団体)によって仕様が定義されているマークアップ言語です。XHTMLの仕様書には、その用途や使用方法が記されていて、記述のルールや文法がDTD(文書型定義)によって厳格に定められています。

最近はXHTMLで作られたウェブページも増えましたが、インターネット上に公開されているウェブページの多くは、文法的に誤ってマークアップされています。この大きな原因は、制作者が正しい書き方を知らないことにあるでしょう。正しい書き方を知っていて、わざと間違っただけのXHTMLを書いているとは考えにくいのですから。

XHTMLのマークアップとは、文書(ページ)のどの部分が「見出し」で、どの部分が「強調」を意味しているとか、どこに「段落」があるかというように、文書を構成するいろいろな要素を明確にするものです。

どこまでを要素として掘り下げるかは、XHTMLの仕様書で定義される範囲となりますが、一般的な文書で使われる要素はほぼカバーされています。逆に言うと、XHTMLで定められている要素のほとんどは、文書を書いていたら普通に使う要素でしかないのです。つまりXHTMLとは、文法を知ってさえいれば、だれにでも正しく

書くことのできるマークアップ言語なのです。

正しく表示され続けること

ブラウザによっては、文法に厳密に従っていないページでもある程度はちゃんと表示してくれます。では、なぜ「正しく」XHTMLを書くのが大切なのでしょう。

XHTML = XML × HTML

XHTML(Extensible HyperText Markup Language)は、ウェブページの作成でおなじみのHTML(HyperText Markup Language)をXML(Extensible Markup Language)のルールに合わせて定義直したものです。HTMLもXHTMLも、人間の言葉で書かれた文書のそれぞれの要素が、文書内でどのような意味を持つかを、それぞれの文法に従って記述する(マークアップする)言語です。XHTMLのマークアップに使う「タグ」はHTMLとほぼ同じですが、

XMLのルールに従うため、微妙な違いがあります。

XHTMLで書かれた文書は、XML文書でもあるため、HTMLのようにただブラウザで表示するだけでなく、XMLのさまざまな派生技術と組み合わせて使うこともできます。

ちなみに、現在主流のXHTMLはバージョン1.1ですが、HTML 4.01を元に作られているので、バージョン番号が若いからといって表現できることが少ないわけではありません。

正しいXHTMLのためのMovable Typeの設定

まず、文法にのっとった正しいXHTMLは、古いバージョンのブラウザでも、最新のブラウザでも、ほとんど正しく表示できます。古いブラウザは、ページの見栄えを調整するCSSという技術に対応していないかもしれませんが、ページに記述された内容を受け取ることは可能です。

HTMLを解釈できるブラウザでさえあれば、文書を構成する要素と、そしてそれらの要素の内容を読み取ってくれるのです。したがって、正しいXHTMLで書かれたウェブページは、多くの人があらゆる手段によって読むことが可能となるでしょう。

また、現在は多少正しくないXHTMLでもブラウザで問題なく表示されますが、将来もずっとそうだとは限りません。しかし正しく書かれたXHTMLならば、今後も正しく表示されると期待できるでしょう。

SEOにも効果を発揮

正しいXHTMLでマークアップされたウェブページは、ブラウザで正しく表示されるだけでなく、それ以外のさまざまなアプリケーションにもメリットをもたらします。

マークアップがされていない文書では、書かれている日本語そのものを、制作者の意図どおりの内容で読み取ってきちんと理解するためには、自然言語の高度な解析処理が必要です。アプリケーションが「見出し」や「強調」がどの部分であるのか、文脈に基づいて判断することは非常に困難です。しかし、正しくマークアップされていれば、どんなアプリケーションでもどこが「見出し」や「強調」なのか一目了然になり、データとして扱って処理を容易に行うことが可能となります。

グーグルやヤフーなどに代表される検索エンジンも、そういったアプリケーションの1つです。それらはtitle要素やh1要素といった、文書のタイトルや大見出しとなる部分を、文脈上、大きな意味を持たせているデータとして扱います。文脈上の「見出し」を正しく「見出し」であるとマークアップしてあればこそ、ウェブページは信用するに足るデータとなるのです。正しいXHTMLはSEO(検索エンジン最適化)にも効果があると言えるでしょう。

XHTMLでマークアップしてみるその前に

HTMLとは異なるXHTMLならではの文法で、いくつか覚えなければならないルールがあります。しかし、ひとまず書き方のサンプルを例示していきますので、それにのっとって書いてみるころから始めてみましょう。ルールを覚えるのはその後でもおおむね大丈夫です。

XHTMLでマークアップをするうえで一番心がけたいことは、「妥当な意味づけ」です。最初は文書に含まれる各パーツが文書内でどういった意味を持つのかを把握して指定できるようになることを目指します。意味づけをすることが当たり前になったら、そこで初めて、XHTMLに関する深い知識を得ていきます。それによって、より良いXHTML文書が書けるようになるでしょう。

……と、実際にマークアップしてみるその前に、Movable Type(以下「MT」)の設定をいくつか変更しておきましょう。

自動XHTML整形を無効にする

[メインメニュー > 編集メニュー > 設定] の [ウェブログの設定] で、[新しいEntryを作る時のデフォルトのテキストフォーマット

ト設定] を「None」にしておきます。これにより、エントリーの作成・編集時の自動整形を標準で無効にできます。記事の作成・編集時に「Convert Line Breaks」に逐次変更することができますが、変更しないでください。

MTの初期設定では、テキストフォーマットは「Convert Line Breaks」になっているので、多くの人はこのテキストフォーマットのまま利用していることでしょう。

「Convert Line Breaks」にしてあると、入力した記事内容が自動的にXHTMLに整形されます。記事テキストで改行されているところへそれぞれ「
」タグが挿入され、空行を段落の境目として、各段落の頭に「<p>」タグが、最後に「</p>」タグが追加されます。このテキストフォーマットのおかげで、入力者はXHTMLを意識せずにそのまま文章を書くだけで、公開されるときにはその文章がXHTMLでマークアップされた状態になるというわけです。

しかし、改行しきれないただの文章では素っ気ないページになってしまうため、通常はリンクを張ったり、箇条書きをしたり、よそのページなどからの引用をしたりといった内容を記事に含めたいでしょう。そのために、段落指定や改行以外のXHTMLタグを記事テキストに書き込むこととなります。



付録のテンプレートを使うブログの管理画面で、ブログの設定 [ウェブログの設定] を表示して、MTでテキストフォーマットに Convert Line Breaks を利用しないように設定しておけば、正しいXHTMLのページを確実に実現できるようになります。

Movable Type が作る「間違ったXHTML」

MTが「Convert Line Breaks」によって行う整形は、XHTMLの文法を理解したうえで行われるのではなく、ただひたすら機械的に改行を「
」タグにして、段落を「<p>」タグと「</p>」タグとで括る、という作業をするだけだという点に注意が必要です。そのために、エントリー内にXHTMLが書かれている場合に、自動整形によって結果として文法違反が発生することがあります。

たとえば、記事内に見出しを付けるときを考えてみましょう。記事内の中見出しは付録のMTテンプレートを使った場合は、

h4要素でマークアップするので、以下のようなXHTMLをテキストエリアに記入することになります。

```
<h4>今日の晩ご飯</h4>
```

「Convert Line Breaks」によって自動整形されると、この部分は公開時には次のようなXHTMLになってしまいます。

```
<p><h4>今日の晩ご飯</h4></p>
```

p要素の中にh4要素が入ってしまっている状態です。XHTMLの文法上、p要素の中にh4要素を入れることはできません。したがって、見出しを付けるためにh4要素

でマークアップするという、正しいことをしていても、結果的には正しくないXHTMLが出力されてしまうというわけです。

せっかく正しいXHTMLの書き方を覚えようとしているのに、これでは逆に間違ったXHTMLを作ってしまう。そこで、自動整形を利用せずに、エントリー本文はすべて自分でXHTMLを完全にマークアップして書き入れていくようにします。そのために、MTの設定でテキストフォーマット「None」を標準にしておくのです。

ただし、「コメントに適用するテキストフォーマットの設定」は「Convert Line Breaks」にしておきます。コメントを書き込んでくれるサイトの読者にまでXHTMLマークアップを求めるわけにはいきません。

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以降で見ると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/>

実習式XHTMLマークアップの「公式」

実習の進め方

付録のMTテンプレートは、エントリーの本文部分以外に関しては妥当な(ルールに従った)XHTMLが生成されるように作られています。あとは、エントリーの内容を自分で正しいマークアップをして書けば、サイト全体として妥当なXHTMLになります。

エントリーに書かれる文章は、いわゆる普通のブログに書く日本語になります。日本語の文章は、多くの場合に実習の各項目で挙げるいくつかの要素に分類できます。この要素とは日本語の文章としての要素ですが、それらには相当するXHTMLの要素があります。そこで、エントリーの内容を要素の分類どおりに分解して、対応するXHTMLでマークアップしていけばいい

のです。

言葉で説明するとややこしく思えるかもしれませんが、そんなに身構えてやるようなことでもありません。とりあえず肩の力を抜いて、120ページ以降の分類された要素の書き方に目を通して予習してください。

ある程度理解したら、XHTMLの要素を考えながら、実際にブログにXHTMLでエントリーを書き込みましょう。

XHTMLマークアップの書き方の例を添えてあります。この例の中で、青色で記された部分は「公式」のようなものだと思います。書き方の例を参考にマークアップする場合は、青色の部分はいじらないで、黒字のところをエントリー本文の該当部分に書き換えればよいでしょう。

XHTMLの要素とタグ、属性

XHTMLは文書構造を定義するものだとも言われます。これは、XHTMLのマークアップ(意味づけ)によって、文書のどこからどこまでが大見出しであるとか、どこが段落になっているかというような構造を記述できるからです。この構造を構成するものを要素(またはエレメント)と呼びます。

“要素”は、開始タグと終了タグによって括られることで、その範囲が明示されます。開始タグは「<」で始まり「>」で終わる文字列で、終了タグは「</」で始まり「>」で終わります。たとえば、body要素は「<body>開始タグ」と「</body>終了タグ」で括られることになります。HTMLでは開始タグや終了タグを省略できる場合もありますが、XHTMLではタグの省略は一切認められていませんので、要素は必ず開始タグと終了タグで括られます。

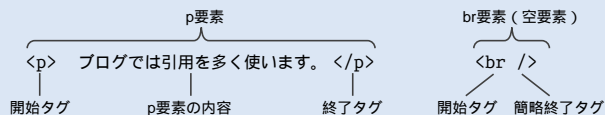
また、HTMLで終了タグを持たなかった(終了タグの記述が文法で禁止されていた)要素のことを空要素と呼びます。たとえばbr要素、img要素、hr要素などです。XHTMLでの空要素は、簡略終了タグというものを利用して、開始タグと終了タグを1つのタグのように一緒に

書いてしまうのが一般的です。br要素は「
」、img要素であれば「」というように書きます。タグの終わりを示す文字列が「/>」になるということです。

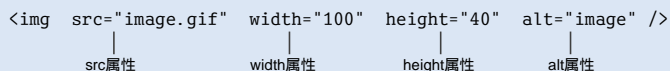
要素によっては属性(またはアトリビュート)を備えていることがあります。属性はその要素の分類や種別を表したり、あるいは要素に情報を付加したりするもので、開始タグの中に記述されます。たとえば画像に使われるimg要素は、画像の所在を示すためのsrc属性や、画像のサイズ情報を付加するためのwidth属性やheight属性といった属性を備えています。

勘違いされやすいのですが、要素とはタグのことではありません。開始タグと終了タグで囲まれている範囲の内容、もしくは空要素のタグが文書構造においてどのような意味合いなのかということです。したがって要素そのものは、いわゆるテキストデータ(文字情報)によって構成されています。ただし、空要素は内容を持ちません。空要素のタグが書かれている部分に、内容の量がゼロの要素が存在していることになります。

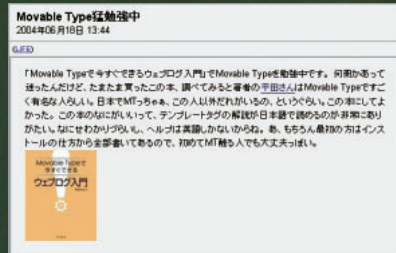
要素には囲む要素と空要素の2種類がある



タグには属性が付くことがある



Convert Line Breaksをオフにした状態でいつものように文章だけ書くと、どれだけ改行を入れていても、下図のように文章がすべてつながって表示されるので注意しよう。



【公式1】: 通常の文章の基本マークアップ < p 要素 >

エントリーの文章は、ほとんどがこの要素によって構成されるでしょう。文章は基本的に p 要素でマークアップします。パラグラフ(段落)になりそうな文章のまとまりを、<p>開始タグと</p>終了タグで括ります。

段落の最後に<p>だけ置くのはXHTMLでは間違った書き方です。また、段落を区切るのに
を使うのはやめましょう。

```
<p>意味のまとまりで段落を区切るのがわかりやすいでしょう。</p>  
<p>段落と段落の間には少し空きができて、読みやすくなります。日本語の文章では行頭を字下げすることがありますが、ウェブでは字下げしないほうが一般的ですね。</p>
```

▼ このXHTMLをブラウザで表示すると.....

意味のまとまりで段落を区切るのがわかりやすいでしょう。

段落と段落の間には少し空きができて、読みやすくなります。日本語の文章では行頭を字下げすることがありますが、ウェブでは字下げしないほうが一般的ですね。

【公式2】: 文章中の強調 < em 要素 >

通常の文章(p要素)の中で、強調したい部分はem要素でマークアップします。

また、タグなどで個別に文字の色や大きさを指定したり、 ~ で太字を指定したりするのもやめましょう。

```
<p>em要素による<em>強調入り</em>のパラグラフ。</p>
```

▼ このXHTMLをブラウザで表示すると.....

em要素による**強調入り**のパラグラフ。

【公式3】: より強い強調 < strong 要素 >

通常の文章(p要素)の中で、より強く強調したい部分を明示するときには、em要素よりもさらに強いstrong要素を使います。

```
<p>strong要素による<strong>さらに強い強調入り</strong>のパラグラフ</p>
```

▼ このXHTMLをブラウザで表示すると.....

strong要素による**さらに強い強調入り**のパラグラフ

✓ DTD(文書型定義)もXHTMLの大切な要素

XHTMLはDTD(文書型定義)によってその文法が厳格に定められています。ある要素の中には別のある要素に入れられるかどうか、ある要素の中にテキストを書けるかどうか、そういったことは文書型定義に書かれています。

たとえばblockquote要素の直下にはテキスト(ブロックレベル要素のタグで囲まれていない文字)を書くことができません。それはDTDが定めているblockquote要素の文法にそぐわないため、書くことと文法違反になるからです。p要素の中にh4要素を入れられないのも同様にDTD

が元になっています。DTDに反した文法で書かれてしまうと、それはXHTML文書ではなくなってしまいます。文法どおりにXHTMLでマークアップしましょう。

XHTML文書の先頭には、通常「<!DOCTYPE」で始まる行があります。これは文書型宣言と呼ばれるもので、その文書では、どのDTDで定め

られたバージョンのXHTMLを使用しているかを宣言するために記述されます。ブラウザもDTDに従ってXHTMLを解釈するので、どのバージョンのDTDに従って書いているのかをXHTML文書中で指定しておくのが大切です。今回使っているのはXHTML 1.1なので、DOCTYPE宣言は次のようになります。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

【公式4】: 引用(ブロックレベル) <blockquote>要素

ブロックレベル(ページ下の囲み参照)での引用はblockquote要素になります。blockquote要素の内容、つまり引用文そのものは、通常の文章(p要素)などのブロックレベルのまとまりとしてマークアップします。blockquote要素のcite属性には、引用元がURLで表せる場合にそのURIを書きます。blockquote要素のcite属性の内容は、Mozillaなどの一部のブラウザでしか表示されませんが、XHTMLとしては備えておくほうが望ましいので「公式」には含めておき、さらに別途cite要素(属性ではなく要素)を指定します(上コード)。

引用元がURLで表せない場合は、「cite=""」のように値が空のcite属性とするのではなく、blockquote要素のcite属性ごと除去してください。また、その場合は、cite要素にある『サイト名「~」から』の記述も、たとえば「読者の感想メールから」のように置き換える必要があります(下コード)。

```
<blockquote cite="http://www.w3.org/">
<div class="quote">
<p>ここには引用された文章が入ります。ここには引用された文章が入ります。ここには引用された文章が入ります。</p>
<!--/quote--></div>
<ul class="cite">
<li><cite>サイト名「<a href="http://www.w3.org/">World Wide Consortium</a>」から</cite></li>
</ul>
</blockquote>
```

▼このXHTMLをブラウザで表示すると.....

ここには引用された文章が入ります。ここには引用された文章が入ります。ここには引用された文章が入ります。

- サイト名「World Wide Consortium」から

```
<blockquote>
<div class="quote">
<p>ここには引用された文章が入ります。ここには引用された文章が入ります。ここには引用された文章が入ります。</p>
<!--/quote--></div>
<ul class="cite">
<li><cite>読者の感想メールから</cite></li>
</ul>
</blockquote>
```

【公式5】: 引用(インライン) <q>要素

通常の文章(p要素)中に、引用した文章が含まれる場合などに利用します。この場合、引用元の明示は、本文内にさりげなく書くといいでしょ(上コード)。

q要素にもcite属性を付けられます。引用元がURLで表せるような場合は、cite属性を付けておくことが望ましいでしょう(下コード)。

```
<p>インラインで引用文を含むパラグラフです。彼は<q>太陽は西から昇るものである</q>と言ったようです。</p>
```

▼このXHTMLをブラウザで表示すると.....

インラインで引用文を含むパラグラフです。彼は 太陽は西から昇るものであると言ったようです。

```
<p>HTML 4.01 は <q cite="http://www.w3.org/TR/html4">HTML 4 is an SGML application conforming to International Standard ISO 8879</q> です。</p>
```

✓ ブロックレベルとインライン

XHTMLの要素は、文書を構成する基本要素である見出しや段落などの「ブロックレベル」の要素と、ブロックレベル要素の中に埋め込まれる形で使われる「インライン」の要素があります。ブロックレベル要素の例としては、h1~h5、p、ul、ol、table、divなどがあります。インライン要素の例としては、a、em、strong、imgなどに加

えて普通の文字があります。ブロックレベル要素の中にインライン要素を含めることはできますが、その逆はできません。インライン要素の中には他のインライン要素を含められます。ブロックレベル要素の中に他のブロック要素を入れられるかどうかは、DTDの定義によります。

【公式6】: 箇条書き < ul 要素・ol 要素 >

単なる箇条書きはul要素で、番号付きの箇条書きはol要素でマークアップします。

箇条書き全体をulまたはolで囲んで、箇条書き項目の1つ1つをli要素でマークアップします。HTMLではを箇条書き項目の先頭に付けるだけでもOKでしたが、XHTMLでは、各項目を正しく ~ で囲みましょう。

箇条書きの項目が1つでも構いません。

```
<ul>
<li>たまねぎ</li>
<li>じゃがいも</li>
<li>にんじん</li>
<li>牛肉</li>
</ul>

<ol>
<li>牛肉を炒めて取っておく</li>
<li>たまねぎを炒める</li>
<li>他の野菜も軽く炒める</li>
<li>材料をスープで煮込む</li>
<li>カレールーを入れる</li>
</ol>
```

▶ このXHTMLをブラウザで表示すると.....

- たまねぎ
 - じゃがいも
 - にんじん
 - 牛肉
1. 牛肉を炒めて取っておく
 2. たまねぎを炒める
 3. 他の野菜も軽く炒める
 4. 材料をスープで煮込む
 5. カレールーを入れる

【公式7】: リンク(インライン) < a 要素 >

通常の文章(p要素)中の任意の文字列をアンカーとしてリンクを張る場合のマークアップです。

HTMLではリンクの<a>要素の属性としてtarget="_blank"のように書くことで、リンク先を別のウィンドウで開くように指定できましたが、XHTML 1.1のa要素にはtarget属性は存在しないので、強制的に別ウィンドウで開くようには指定できません。閲覧者自身が別ウィンドウを開きたいときに、ブラウザの機能を使って任意に別ウィンドウを開くように操作してもらうこととなります。

```
<p>文中の<a href="http://www.impress.co.jp/">任意の文字列</a>でリンクする。</p>
```

▼ このXHTMLをブラウザで表示すると.....

文中の 任意の文字列 でリンクする。

【公式8】: リンクリスト < ul 要素 + a 要素 >

リンクを示すアンカーを通常の文章(p要素)から外に出して、リンク先の一覧を明示するような場合のマークアップです。

本文で参照しているリンク先が多数ある場合や、長い文章で関連サイトを最後にまとめる場合などに使うといいでしょう。

リスト項目が1つでも構いません。

こちらにもインライン要素のリンクの場合と同様に、a要素にtarget属性を書くことはできません。

```
<ul class="linklist">
<li><a href="http://w3j.org/">W3J.org</a></li>
<li><a href="http://altba.com/yuu/blog/">securecatのMT</a></li>
</ul>
```

▼ このXHTMLをブラウザで表示すると.....

- W3J.org
- securecatのMT

【公式9】：整形済みテキスト <pre> 要素 >

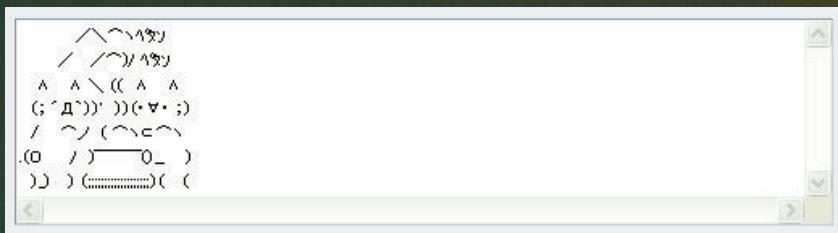
XHTMLでアスキーアートを簡単に表現するには、pre要素を使います。pre要素は改行や空白が、書いたとおりにそのまま表示される「整形済みテキスト」を指定するものです。

ただし、フォントセットの関係で、OSやブラウザの設定によってはうまく表示されないこともあります。

付録のStylesheetテンプレートでは、pre要素のボックス幅がコンテンツエリアの幅を超える場合はスクロールバーが出るようにしていますが、バグ対策のためにMac版のIE 5ではスクロールバーが表示されません。pre要素の内容は適宜改行して1行が長くないようにするのが無難です。

```
<pre>        /\  \ `ベツ
      /  /  )ベツ
    _  \ ( ( _
   (;  ` `))') )( ` ` ;)
  /    \  ( ` ` \ `
.(0    / )   ( )_   )
  ) ) ( ; ; ; ; ; ; ; ; ; ; ; ) _ (</pre>
```

▼このXHTMLをブラウザで表示すると.....



【公式10】：見出し <h4> 要素・<h5> 要素 >

記事を書くときに入力した「タイトル」は、エントリーの大見出しとなりますが、エントリー中で、さらに中見出し、小見出しというように見出しを付けると、エントリーの構成を見やすく整えられます。文章が長くなる場合などに有効でしょう。

エントリー内の中見出しはh4要素で、小見出しはh5要素でマークアップします。

普通は見出しはh1要素から始まりますが、ブログのタイトルにh1要素を、日付にh2要素を、エントリーの大見出しにh3要素を使っているので、エントリー内ではh4要素から始めることとなります。

```
<h4>中見出し</h4>
<p>中見出しに続く通常の文章です。中見出しに続く通常の文章です。</p>

<h5>小見出し</h5>
<p>小見出しに続く通常の文章です。小見出しに続く通常の文章です。</p>
```

▼このXHTMLをブラウザで表示すると.....

中見出し

中見出しに続く通常の文章です。中見出しに続く通常の文章です。

小見出し

小見出しに続く通常の文章です。小見出しに続く通常の文章です。



後編につづく

これで森田講師による「正しいXHTML塾」前編を終わります。後編では、同じMTテンプレートを使ったさらに高度なXHTML記述の公式と、具体的にブログで記事を書く際の書き方を紹介します。後編で扱う内容には、内容の追加や削除、ルビ(ふりがな)、プログラムコ

ード、画像、画像のテキスト回り込み、特殊記号などが含まれます。

後編は2004年9月29日発売予定のインターネットマガジン 11月号に掲載予定です。お楽しみに。



[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