

さらばタグ

HTML 4.0とCSS 2によって、「ブラウザ戦争」によるタグの拡張の時代は終わった。これからはIEやナビゲーターでの見え方ではなく、規格に従ったHTMLの作成が求められる。今こそホームページのスタイルシート化を始めよう。

ホームページ スタイルシート化作戦

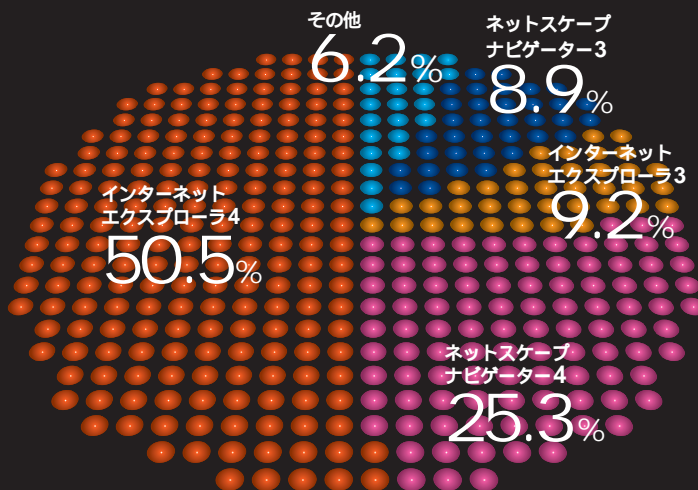
スタイルシート対応のブラウザが8割に

右のグラフは、今年9月3日から5日までの3日間、インターネットマガジンのホームページ (<http://internet.impress.co.jp/>)を訪れたブラウザを集計したものだ。スタイルシートをサポートしたインターネットエクスプローラ (IE) 3、4とネットスケープナビゲーター4の割合が合わせて約85%に達している。古いバージョンのブラウザを考慮して、スタイルシートを使うことを今までためらっていた人も多いただろう。しかし、この結果を見ると、スタイルシートを活用したページ作りがようやく実用段階に入ったと言える。

スタイルシートを使うときのポイントは、新機能を使った派手なページを作るのではなく、「構造」と「見栄え」を分離することだ。HTMLでページの骨格を作り、飾りはスタイルシートで付ければ、

古いブラウザでも構造だけはきちんと見えるページが作れる。このポイントにもとづいて、ホームページをスタイルシート化していくための実践的な手順を紹介していこう。

なお、この記事では、スタイルシートのすべての機能やブラウザ間の違いを細かく解説しているわけではない。詳しくは本誌付録の小冊子「HTML4.0 & スタイルシート完全互換性データファイル」を参照してほしい。



【HTML 4.0】

W3C (World Wide Web Consortium) が定めたHTMLの最新の標準規格。文書の構造と見栄えを明確に分離することが基本となり、見栄えにはスタイルシートを使うことが推奨されている。

【CSS】カスケードリングスタイルシート

いくつかあるスタイルシートの中で、もっともよく使われ、W3Cでも規格化されているスタイルシート。1996年12月に発表されたCSS1と1998年5月に発表されたCSS2がある。IE4、ナビゲーター4ともに、不完全ながらCSS1のほとんどをサポートし、CSS2の一部を先取りしている。この記事で「スタイルシート」と呼んでいるのは、CSSのことである。

5つの利点

スタイルシート

H TMLが簡単になる

「スタイルシートだと新しいことを覚えなきゃならなくて面倒くさい」と思っているだろうか。確かにスタイルシートは一見ややこしい英単語の羅列のようだ。しかし、スタイルシートを使って「構造」と「見栄え」を分ければ、HTMLはずっと簡単になる。ほんの初歩的なタグを数個使ってHTMLを書いておけば、後からいくらでもスタイルシートでデザインを加えることができる。タグを複雑に組み合わせてトリッキーなHTMLを作る必要はない。

だ れでも読めるページが作れる

WWWは、「パソコン+WWWブラウザ」という組み合わせのためにあるわけではない。今後、携帯端末、音声読み上げソフト、点字出力など、さまざまな環境での利用が進んでいくだろう。CSS2では、「メディアタイプ」というしくみが導入された。まだサポートしているソフトウェアはないが、メディアタイプが使われるようになれば、右のように、1つのHTMLに対していろいろな環境に合わせたスタイルシートを指定できるようになる。

```
@media aural {
  音声の場合のスタイル.....
}
@media braille {
  点字の場合のスタイル.....
}
@media handheld {
  小さな画面のためのスタイル.....
}
@media screen {
  パソコンの場合のスタイル.....
}
```

豊 かな表現ができる

もともと文書の構造だけを記述するものだったHTMLでデザインまでしようとする、どうしても中途半端になってしまう。スタイルシートを使えば、ポイント数で文字の大きさを決めることもできるし、文章の上下左右の余白を変えることもできる。CSS2の「ポジショニング」を使えば、好きなところに文字や画像を配置することもできる。しかもスタイルシートはテキストだけで書ける。レイアウトのために<TABLE>を使ったり、画像だらけのページにしなくてもすむ。

繰 り返しの手間が省ける

ホームページ作成ソフトに頼る人が多いのは、作業をすばやく行いたいからだ。ページ中の文字の色を変えるためにいちいち「」と何度も入力するのは確かに面倒だ。スタイルシートの利点はこうした繰り返しの手間を省く点にもある。下の例のように「見出しはこの色」と1回指定するだけで、1つのページの見出しの色をすべて変更できる。複数のページの見出しを一気に変更することも可能だ。

```
<H1><FONT COLOR="#FF0000">赤い見出し</FONT></H1>
<H1><FONT COLOR="#FF0000">また赤い見出し</FONT></H1>
<H1><FONT COLOR="#FF0000">ここも赤い見出し</FONT></H1>
```

```
H1 { color: #FF0000; }
+
<H1> 赤い見出し</H1>
<H1> また赤い見出し</H1>
<H1> ここも赤い見出し</H1>
```

将 来のWWWに対応できる

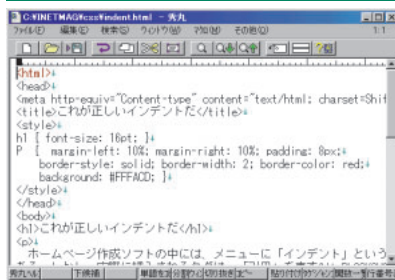
これからのWWWで広く使われるだろうと期待されているのがXMLというマークアップ言語だ。XMLを使えば、自由に文書やデータのためのタグを新しく作ることができる。しかし、XMLは文書やデータの構造だけを記述するもので、「どう表示されるか」については何も指定できない。XMLの表示を決めるのはスタイルシートだ。HTMLでは文書の構造だけを書き、スタイルシートでデザインを作るという習慣を今からつけておけば、XMLが普及する時代にもついていけるだろう。

スタイルシート化の準備をしよう

1 テキストエディターを用意する

HTMLを書くには、それなりのソフトウェアが必要だと思っている人もいるかもしれない。しかし、現在のホームページ作成ソフトでは、「構造」と「見栄え」を明確に区別できず、自分の意図しないタグが挿入されてしまう。ワープロソフトで作ると、タグだらけのHTMLができあがる。スタイルシート作成を完璧にサポートしたソフトウェアはまだ存在しないのが現状だ。

スタイルシート化したページを作るなら、テキストエディターで直接HTMLを書くのがベストだ。「メモ帳」などOS付属のものでもいいが、どうせなら高機能なシェアウェアを用意したい。ウィンドウズなら、老舗のテキストエディター「秀丸エディタ」がおすすめだ。強力なマクロ機能を持ち、HTMLをサポートしたマクロが有志の手で公開されている。マッキントッシュなら「Jedit」で決まりだろう。マクロが最初から付属しており、メニューからHTMLのタグを挿入することもできる。



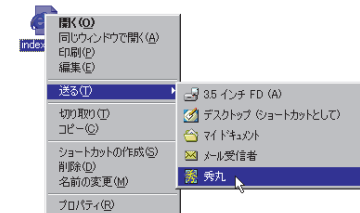
URL <http://hidemaru.xaxon.co.jp/index.html>
CD-ROM収録先 Win Hidemaru
シェアウェア 4,000円



URL <http://www.ijinet.or.jp/matsumoto/>
CD-ROM収録先 Mac Jedit
シェアウェア 2,500円

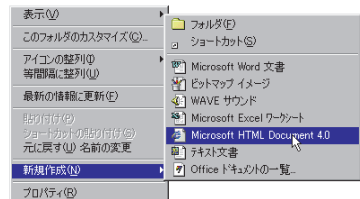
右クリックでHTMLを編集

ウィンドウズでHTMLを編集するときには、フォルダー上での右クリックメニューを最大限に活用したい。テキストエディターのインストーラーに「送るに入れる」という項目があれば、チェックしておこう。ファイルの右クリックで出るメニューの「送る」からエディターを選べば、即座にHTMLが編集できる。



右クリックでHTMLを作成

ウィンドウズのレジストリーの操作に自信のある人なら、フォルダーを右クリックしてHTMLを作成できるようにしてみよう。「レジストリエディタ」を使って「HKEY_CLASSES_ROOT」「.html」の下に「ShellNew」というキーを作り、そのキーの下に名前が「FileName」、データが「new.html」という文字列を作る。次にテンプレートとなるHTMLファイルを作り、「new.html」という名前を付ける。これをエクスプローラでWindowsフォルダーの下の「ShellNew」フォルダーにコピーする。これで、右クリックメニューの「新規作成」からHTMLファイルを作れるようになる。



2 シンプルなHTMLを作る

テキストエディターでHTMLを書くからといって、参考書と首っ引きですべてのタグを覚える必要はない。<HTML>、<HEAD>、<TITLE>、<BODY>といった必須のタグのほか、<H1> ~ <H6>（見出し）、<P>（段落）、
（改行）、<A>（リンク）だけ覚えれば立派なページが作れる。画像を貼ったりプラグインを使ったりすることは最後に考えればよい。まずはテキスト中心のシンプルなページを作ろう。

ホームページ作成ソフトで作ったページがすでにあるなら、テキストエディターで開いて、余計なタグをどんどん消してしまおう。文字の大きさや色のためのタグや、インデントのための<BLOCKQUOTE>、<TABLE>タグはもう必要ない。

シンプルなHTMLができれば、いよいよスタイルシートの出番だ。

```
<HTML><HEAD>
<TITLE>古いHTML</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF"
TEXT="#000000">
<CENTER>
<FONT SIZE=5 color="#FF0000">
<B>見出し</B></FONT>
</CENTER>
<BLOCKQUOTE>本文.....
```



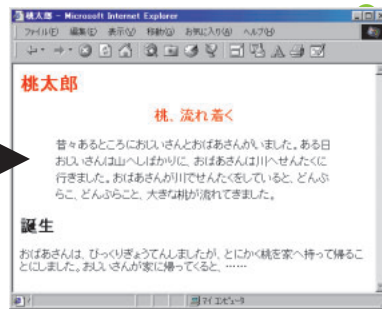
```
<HTML><HEAD>
<TITLE>新しいHTML</TITLE>
</HEAD>
<BODY>
<H1>見出し</H1>
<P>本文.....
```

スタイルシート化への3つのステップ

1 タグにSTYLE属性を付けてみる

まず、スタイルシートの効果を簡単に見るために、右のサンプルのように<H1>や<H2>、<P>などのタグの中にSTYLE属性を付けてみよう。「STYLE=" "」の「" "」の中に、「プロパティ: 値; プロパティ: 値;」のように「:」でプロパティと値をペアにし、「;」で区切って並べていく。右のサンプルでは、「color」や「font-size」がプロパティで、「red」や「16pt」が値だ。ただし、STYLE属性を使う方法は、効果をテストする場合などにとどめ、以下の2や3のようにスタイルをまとめて書くのが常道だ。

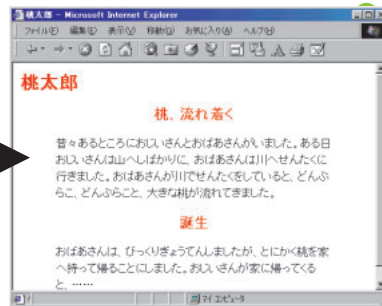
```
<HTML>
<HEAD><TITLE>桃太郎</TITLE>
</HEAD>
<H1 STYLE="color: red; font-size: 20pt;">桃太郎</H1>
<H2 STYLE="color: red; font-size: 16pt; text-align: center;">桃、流れ着</H2>
<P STYLE="margin: 16px 10%; line-height: 150%;">昔々あるところに...
```



2 <STYLE> タグでまとめる

STYLE属性でスタイルシートの効果がつかめたら、今度はHTMLの先頭の<HEAD> ~ </HEAD>の中に<STYLE>タグを作成しよう。<STYLE> ~ </STYLE>の間には、右のサンプルのようにタグ名（「H1」など）の後ろに「{ }」を書く。かつこの中に1で作ったSTYLE属性の内容を移し、STYLE属性は消してしまおう。これで、HTML内のすべての<H1>、<H2>、<P>タグに指定したスタイルが反映される。なお、<STYLE>タグの中は、スタイルシートに対応していないブラウザで表示されないように<!-- ~ -->のコメントで囲うこと。

```
<HTML>
<HEAD><TITLE>桃太郎</TITLE>
<STYLE>
<!--
H1 { color: red; font-size: 20pt; }
H2 { color: red; font-size: 16pt; text-align: center; }
P { margin: 16px 10%; line-height: 150%; }
-->
</STYLE>
</HEAD>
```



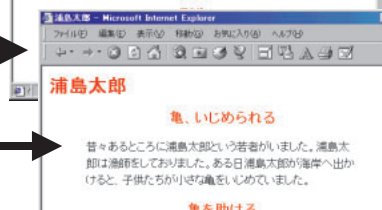
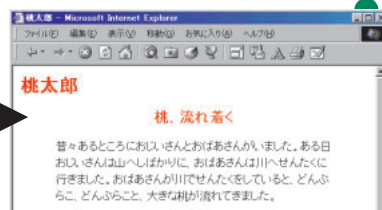
3 CSSファイルを作る

STYLE属性や<STYLE>タグを使っただけでは、スタイルシートの力を100%引き出すことはできない。スタイルシートをHTMLファイルの外に置き、複数のHTMLファイルでスタイルを共有してこそ、スタイルシートの本当の価値が出る。まず、2で作った<STYLE>タグの内容を別のテキストファイルにコピーし、ファイルの拡張子を「css」に変える。次に、そのスタイルを反映させたいHTMLファイルの<HEAD> ~ </HEAD>の中に、<STYLE>タグの代わりに右のソースのように<LINK>タグを書いてHTMLファイルとCSSファイルを結び付ける。また別のHTMLファイルにもこの<LINK>タグを1行書けば、同じCSSファイルのスタイルを読み込むようになる。

```
mukasi.css
H1 { color: red; font-size: 20pt; }
H2 { color: red; font-size: 16pt; text-align: center; }
P { margin: 16px 10%; line-height: 150%; }
```

```
momo.html
<HTML>
<HEAD><TITLE>桃太郎</TITLE>
<LINK REL="stylesheet" TYPE="text/css" HREF="mukasi.css">
</HEAD>
```

```
urasima.html
<HTML>
<HEAD><TITLE>浦島太郎</TITLE>
<LINK REL="stylesheet" TYPE="text/css" HREF="mukasi.css">
</HEAD>
```



色と背景を指定しよう



EM { color: red; }

<BODY BGCOLOR="#FFC0CB">



BODY { background: #FFC0CB; }

スタイルシート化の手始めに、色と背景を指定する方法を覚えよう。文字の色を指定するプロパティは「color」だ。ほとんどどんなタグにでも指定できる。

「color」には、HTMLで色を指定する場合と同じく、「#FF0000」と16進数で指定したり「red」と色名で指定したりできるが、次のようにRGB 3原色で指定する方法を使うとわかりやすいだろう。

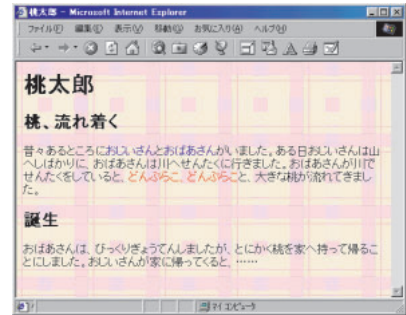
```
color: rgb(255, 0, 0);
color: rgb(100%, 0%, 0%);
```

前のページまで読めば、見出しや段落全体の色を指定する方法はわかるが、文章の中の特定の文字だけ色を付けたいときはどうすればいいだろうか？ここで活躍するのが、今まであまり使われてこなかったやといった「強調」を意味するタグだ。スタイルシートで「EM { color: red; }」とすれば、 ~ で囲った部分はすべて赤になる。で何度も色を指定しなくてもいいし、色を変えたいときにもスタイルシートを1か所書き換えるだけですむ。スタイルシートに対応していないブラウザでも、太字や斜体で表示されるのが利点だ。

「この文字に色を付けたいが、強調の意味ではない」という場合は、を使おう。「SPAN.クラス名」の形でスタイルを指定し（下のサンプルでは「person」がクラス名）、HTMLの中では色を付けたい文字を ~ で囲む。クラス名を使ったスタイル指定をすれば、特定の個所のスタイルを変えられるだけではなく、「person」のようにタグに自分で意味を付けることができる。

背景を指定するプロパティにはいくつかあるが、色と画像の指定だけ覚えればいだろう。背景色は、「background」に「color」と同じ方法で色を指定する。背景画像は、「background: url(back.gif);」のように「url()」の「()」内にファイル名を指定する。「url(../img/back.gif)」のように相対パスや絶対パスでも指定できる。色と画像の両方を指定するには、「background: url(back.gif) #FFFFFF0;」とする。

背景色や背景画像といえば思い付くのが<BODY>と<TABLE>だが、スタイルシートを使えば、そのほかのタグにも背景を指定できる。<P>に背景色を指定すれば、背景色のためだけに<TABLE>を使う必要はなくなる。



注意点

IE3では「rgb()」形式での色の指定は効かない。
CSSでは色名は「red」や「blue」などの基本16色だけで、IEやナビゲーターの「lightblue」や「pink」などは規則にない。
ナビゲーター4では、<H1>や<P>に背景色を指定すると文字の背景にだけ色が付く。背景全体に色を塗るには枠線を指定する。P.297参照。



```
<BODY BGCOLOR="#FFC0CB" TEXT="#000000"
BACKGROUND="back.gif">
<P>昔々あるところに
<FONT COLOR="#0000FF">おじいさん</FONT>と
<FONT COLOR="#0000FF">おばあさん</FONT>が
いました。.....川でせんとくをしていると、
<FONT COLOR="#FF0000">どんぶらこ、
どんぶらこ</FONT>と、大きな桃が流れてきました。
```



```
BODY { color: black;
background: #FFC0CB url(back.gif) }
EM { color: red; font-style: normal; }
SPAN.person { color: blue; }
```



```
<BODY>
<P>昔々あるところに
<SPAN class="person">おじいさん</SPAN>と
<SPAN class="person">おばあさん</SPAN>が
いました。.....川でせんとくをしていると、
<EM>どんぶらこ、どんぶらこ</EM>と、
大きな桃が流れてきました。
```



フォントを使いこなせ



BIG { font-size : 20pt; }



P { font-family:'MS P 明朝','平成明朝',serif; }

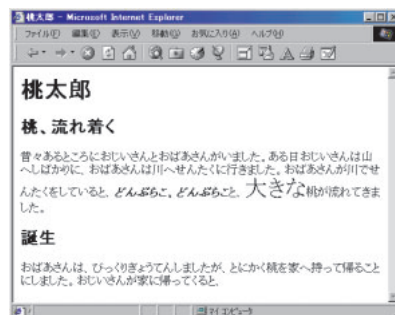
は文字の大きさを変えたり色を付けたりにするために頻繁に使われているタグだ。しかし、HTML 4.0ではをはじめとする文字装飾のためのタグは「不適切」なものとして廃止される方向に進んでいる。スタイルシートを使えばタグが不要になるばかりでなく、ポイントやピクセル単位でサイズを指定できるので、さらに便利になる。たとえば「font-size: 48pt;」などとして、目を引くタイトルを作ることも可能だ

文章の途中で文字のサイズを変えたいときには、左ページで説明したようにやを使うほかに、<BIG>や<SMALL>にスタイルシートでサイズを指定して使う方法がある。<BIG>と<SMALL>はHTML 4.0でも廃止予定になっていないので積極的に活用しよう。

フォントの種類、つまりゴシック体や明朝体

を指定するには、「font-family」を使う。「MS P 明朝」など自分のパソコンにあるフォントをついで使ってしまうが、読む人の環境に同名のフォントがなければ意図どおりに表示されない。おすすめはOSや言語に依存しない汎用フォント名も指定しておくことだ。指定したフォントがなくても日本語のページでは「serif;」を追加すれば明朝体に、「sans-serif;」を追加すればゴシック体になる。

太字や斜体にするときは、や<I>を使用してかまわない(HTML 4.0では廃止予定ではない)。ただし、特定のタグをまとめて太字にしたり斜体にしたりするには、スタイルシートを使う必要がある。スタイルシートでの太字の指定は「font-weight: bold;」、斜体は「font-style: italic;」だ。逆に太字や斜体にしないときは、値に「normal」を指定する。



注意点

ネットスケープナビゲーター4では、日本語のページで「font-family」を指定しても反映されないことが多い。



.....おばあさんが川でせんたくをしていると、
<I> どんぶらこ、どんぶらこ</I>と、
大きな桃が流れて
きました。



P { font-family:'MS P 明朝','平成明朝',serif; }
BIG { font-size: 24pt; }
EM { font-weight: bold; font-style: italic; }



<P>.....おばあさんが川でせんたくをしていると、
どんぶらこ、どんぶらこと、
<BIG>大きな</BIG>桃が流れてきました。

POINT H1はフォント用のタグではない

HTMLの入門書には、「<H1> ~ <H6>を使うとフォントのサイズが変わります」と書いてあるものが多いが、それは間違いだ。IEやナビゲーターで文字の大きさが変わるとしても、それはブラウザが大きさを決めているだけで、「たまたま」そうなっていると考えるべきだ。<H1> ~ <H6>以外に間違いやすいタグには、次のようなものがある。

タグ	正しい意味	間違い
<P>	段落	1行空ける
<BLOCKQUOTE>	引用	インデント
<ADDRESS>	連絡先	斜体
	強調	斜体
	強い強調	太字

また、(リスト)や<DL>(定義リスト)をインデントのために使うのも避けよう。スタイルシートを使わずに今までどおりHTMLだけで書きたい場合には、文字の大きさや色を変えるためにタグを使ってもかまわないが、<BLOCKQUOTE>など「構造」を表すタグを「見栄え」のために使ってはならない。

文字揃えと行間を指定しよう

<code><CENTER></code>	➔	<code>DIV.center { text-align: center; }</code>
<code><H2 ALIGN="right"></code>	➔	<code>H2 { text-align: right; }</code>

スタイルシートを使うのは、飾りを付けるためだけでなく、テキストを読みやすくするためでもある。ここでは文字揃えと行間の指定について説明しよう。

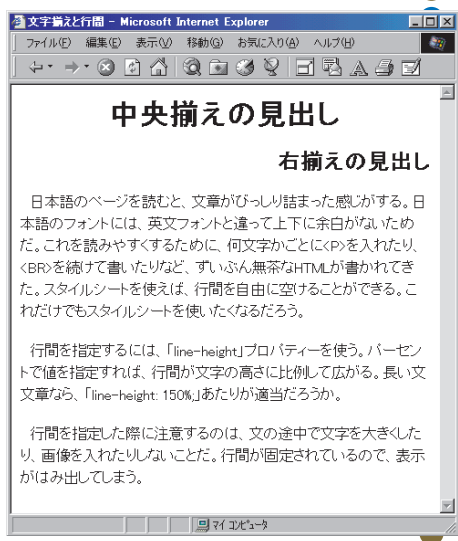
HTMLのタグと言えば、`<H1>`や`<P>`よりも`<CENTER>`が先に思い浮かぶ人もいるかもしれない。しかし、`<CENTER>`はもちろん、`<H1>`や`<P>`タグのALIGN属性もHTML 4.0では廃止予定となっている。

スタイルシートで文字揃えを指定するには、「text-align」プロパティを使う。値には、「left」、「right」、「center」が指定できる。`<CENTER>`タグを使う代わりにページの特定の部分をまとめて中央揃えにしたいければ、クラス指定を使って「`DIV.center { text-align: center; }`」とし、中央揃えにしたい部分を`<DIV class="center"> ~ </DIV>`で囲めばよいだろう。

日本語のページを読むと、文章がびっしり詰まった感じがする。日本語のフォントには、英文フォントと違って上下に余白がないためだ。これを読みやすくするために、何文字かごとくに`<P>`を入れたり、`
`を続けて書いたりなど、ずいぶん無茶なHTMLが書かれてきた。スタイルシートを使えば、行間を自由に空けることができる。これだけでもスタイルシートを使いやすくなるだろう。

行間を指定するには、「line-height」プロパティを使う。パーセントで値を指定すれば、行間が文字の高さに比例して広がる。長い文章なら、「line-height: 150%;」あたりが適当だろうか。

行間を指定した際に注意するのは、文の途中で文字を大きくしたり、画像を入れたりしないことだ。行間が固定されているので、表示がはみ出してしまふ。



18 `<CENTER><H1>` 中央揃えの見出し
`</H1></CENTER>`
`<H2 ALIGN="right">` 右揃えの見出し `</H2>`
 日本語のページを読むと.....

➔ `H1 { text-align: center; }`
`H2 { text-align: right; }`
`P { line-height: 150%; }`

新

➔ `<H1>` 中央揃えの見出し `</H1>`
`<H2>` 右揃えの見出し `</H2>`
`<P>` 日本語のページを読むと.....

POINT 段組みはあきらめよう

「このページは800 x 600以上の画面でごらんください」と書いてあるページを見かけるが、ページを作るときには、画面の大きさにこだわる必要はない。訪れる人はPDAの狭い画面で見ているかもしれないし、音声読み上げソフトなど「画面のない」環境なのかもしれない。「相手が何を使っているかなんてわかりようがない」と考えたほうがいい。画面の大きさを指定しなくなってしまうの

は、`<TABLE>`を使って「段組み」を表現するページを作った場合だろう。しかし、`<TABLE>`は「表」のためのタグであり、HTMLでもスタイルシートでも「段組み」を表すことはできない(ナビゲーターの独自タグに`<MULTICOL>`があるが、IEではサポートされていない)。無理をしてHTMLの規則にはずれた段組みを作っても、すべての環境できれいに見えずと

は限らない。IEやナビゲーターの場合でも、大きな表は表示するまで時間がかかり、読む人に負担を与えてしまうことになる。正しいHTMLを書きたいなら、段組みはあきらめよう。どうしても左右の方向にレイアウトしたいなら、P.298のようにスタイルシートの「ポジショニング」を使おう。

これが正しいインデントだ

```
<BLOCKQUOTE>
<CENTER><TABLE width="80%">
```



```
P{margin-left: 10%; margin-right: 10%; }
```

ホームページ作成ソフトの中には、メニューに「インデント」という項目のあるものがある。しかし、実際に挿入されるタグは、「引用」を表す<BLOCKQUOTE>だ。また、<TABLE>を使ってインデントを付けているページも多い。文章の左右に余白を空けて読みやすくしたいと思うのはもっともだが、意味の間違ったHTMLを書くのは避けたいものだ。余白を空けるにはスタイルシートを使う。

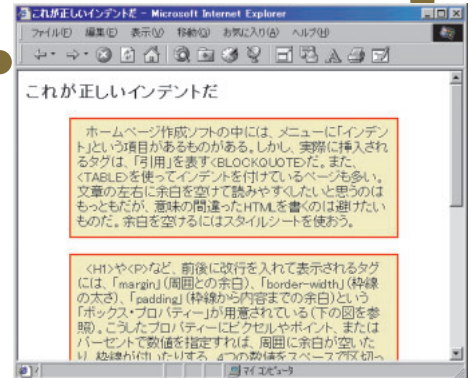
<H1>や<P>など、前後に改行を入れて表示されるタグには、「margin」（周囲との余白）、「border-width」（枠線の太さ）、「padding」（枠線から内容までの余白）という「ボックス・プロパティ」が用意されている（下の図を参照）。こうしたプロパティにピクセルやポイント、またはパーセントで数値を指定すれば、周囲に余白が空いたり、枠線が付いたりする。4つの数値をスペースで区切

って並べると、順に上、右、下、左の値になる。2つの数値なら上下と左右の値になる。段落の上に8ピクセル、右にページの10%、下に16ピクセル、左にページの20%の余白を空けるには、次のように設定する。

```
P { margin: 8px 10% 16px 20%; }
```

上下左右どれかの大きさだけを指定したい場合は、「margin-left」のように、各プロパティに「-left」、「-right」、「-top」、「-bottom」を付けて指定する。枠線の場合は、「border-left-width」のようにする。

さらに、「background」で背景色を付けたり、「border-color」で枠線に色を付けたりすれば、背景や枠線のためにわざわざ<TABLE>を使う必要がなくなり、正しいHTMLのまま自由なデザインが可能だ。



注意点

枠線を表示させるには、太さの指定と同時に「border-style: solid;」も指定する。IE 3でも効果があるのは、「margin-left」、「margin-right」、「margin-bottom」のみ。



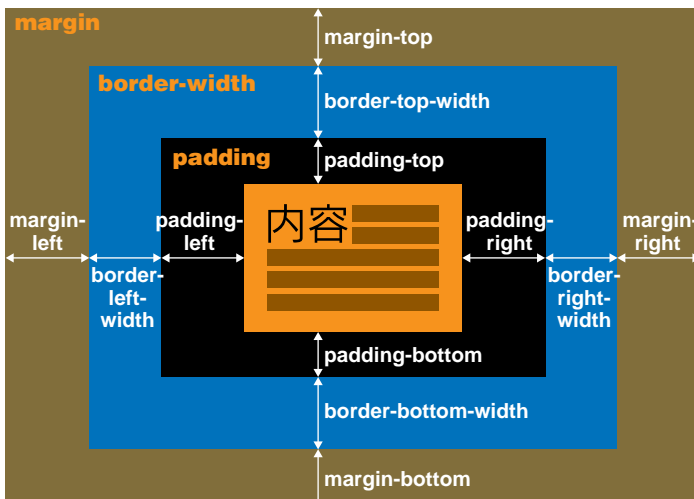
```
<CENTER>
<TABLE BORDER=2 BORDERCOLOR="red"
  BGCOLOR="#FFFACD" WIDTH="80%">
<TR><TD> ホームページ作成ソフトの.....
</TD></TR></TABLE>
```



```
P{ margin-left: 10%; margin-right: 10%; padding: 8px; border-style: solid; border-width: 2; border-color: red; background: #FFFACD; }
```



```
<P> ホームページ作成ソフトの..... </P>
```



そのほかの枠線の使い方

枠線のスタイルは、「border」でまとめて指定できる。

```
border-style: solid; border-width: 2px;
border-color: red;
```

```
border: solid 2px red;
```

また、IE 4では、「border-color」で枠線の色を上下左右別々に指定できる。

```
border-color: red blue green yellow;
```

太さを指定するときと同じく、順に上、右、下、左の色になる。

表組みを使わずにレイアウト

<TABLE>



DIV { position: absolute; }

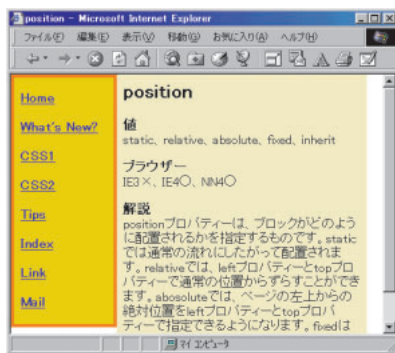
<TABLE>タグはナビゲーター1.1で登場したが、当初は純粋に「表」を作るために使われていた。やがて「段組み」のために使われるようになり、現在ではページ全体のレイアウトに使われるようになってしまった。現在のHTMLの「構造」と「見栄え」の混乱を代表するタグと言える。

CSS2では、文字や画像の位置を自由に指定するための「ポジショニング」が用意され、<TABLE>タグを使わなくても、柔軟なページデザインができるようになった。すでにIE4とナビゲーター4にも実装されている。ナビゲーター4の独自タグ<LAYER>でも位置を指定できるが、IE4でも使えるスタイルシートを利用するほうがよいだろう。

ポジショニングを使うには、<DIV>などにクラス名を付け（サンプルでは「DIV.left」など）、「position: absolute;」を指定する。次に「left」プロパティと「top」プロパティ

でページの左からの位置と上からの位置を指定する。「width」プロパティと「height」プロパティを指定すれば、横と縦の長さも指定できる。HTMLの中で配置したい部分を<DIV CLASS="left"> ~ </DIV>で囲えば、指定した位置に表示されるようになる。

サンプルは、ポジショニングを使って左にナ



ビゲーション用のリンクを、右に本文を置いたありがちなデザインのページだ。現在こうしたページは<TABLE>で作られているが、これからはスタイルシートで置き換えることを検討するべきだろう。

もちろんスタイルシートをサポートしていないブラウザでは位置の指定は反映されないが、上から順番に表示されるだけなので、意味も表示も問題のないHTMLになる。

注意点

IE3では、ポジショニングは使えない。ナビゲーター4で「position」を指定した場合、ウィンドウをリサイズするとスタイルが崩れることが多い。

```
<TABLE CELLSPACING=8><TR>
<TD WIDTH=130 BGCOLOR=" #FFD000">
<A HREF=".....">Home</A>..... </TD>
<TD BGCOLOR="#FFF8CD"><H1>position</H1>.....
</TD></TR></TABLE>
```



```
DIV.left { position: absolute; left: 0; top: 0; width: 130px;
padding: 8px; border: solid 4px #FF8000;
background: #FFD000; font-weight: bold; }
DIV.right { position: absolute; left: 130px; top: 0px;
background: #FFF8CD; padding: 8px; }
```



```
<DIV class="left"><A HREF=".....">Home</A>.....
</DIV>
<DIV class="right"><H1>position</H1>..... </DIV>
```

ブラウザーごとにスタイルシートを切り替える

スタイルシートを使う際に悩まされるのが、ブラウザーの違いによって、画面の表示が大きく乱れる場合があることだ。特にスタイルシートの一部しかサポートしていないIE3や、スタイルシートの実装がやや弱いナビゲーター4が問題になる。

そこで、スタイルシートを複数用意して、ブラウザーによって切り替えるテクニックを紹介しよう。スクリプトを使ってブラウザーの種類を判別し、<LINK>タグ内のCSSファイルの名前を自動的に変えればよい。ブラウザーを表す「navigator.userAgent」に含まれる

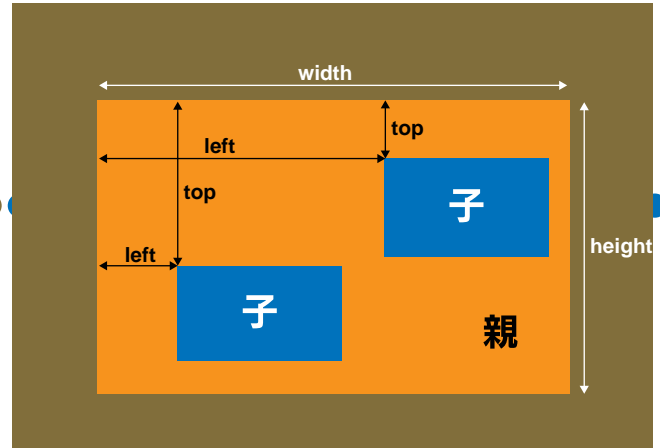
文字で場合分けをして、<LINK>タグの文字列（以下のソースでは「tag」）を組み立てる。でき上がったタグを「document.write」でHTMLに書き込むだけで。

```
<SCRIPT>
var ua = navigator.userAgent;
var tag = '<LINK REL="stylesheet" TYPE="text/css" HREF=""';
if(ua.indexOf("MSIE 4") >= 0) tag += 'ie4.css';
else if(ua.indexOf("Mozilla/4") >= 0) tag += 'nn4.css';
else if(ua.indexOf("MSIE 3") >= 0) tag += 'ie3.css';
else tag += 'other.css';
tag += '>';
document.write(tag);
</SCRIPT>
```

以下のソースを<HEAD> ~ </HEAD>タグに入れて、「ie4.css」、「nn4.css」、「ie3.css」、「other.css」の4つのCSSファイルを用意してみよう。

新

さらに高度なレイアウトに挑戦



前のページで紹介したポジショニングは、ページ全体の中での位置を指定する方法だ。ポジショニングをさらに応用して、右の図のように特定の親ブロックの中で子の位置を指定し、親ブロックの前後では文字が普通に流れるようにすれば、イメージマップのような表現になる。これはIE4でしかうまくいかないが、将来を見すえたページ作成をするなら、一度トライしておくのもいいだろう。

まず、親ブロックとなる<DIV>タグのスタイルを作り、クラス名(ここでは「map」)を付

ける。「position: relative;」を指定し、「width」プロパティと「height」プロパティで横と縦の長さを指定しておく。これで<DIV>タグはほかの文と並んで上から順に表示されるが、その内部では文字や画像を自由に配置できるようになる。<DIV>タグの背景に画像や色を指定しても面白い。

```
1 DIV.map { position: relative; width:320; height: 240;
  background: #FFA500; font-weight: bold;
  font-size: 20pt; font-family: Verdana; }
  DIV.link1 { position: absolute; left: 150; top: 20; }
  DIV.link2 { position: absolute; left: 20; top: 65; }
  DIV.link3 { position: absolute; left: 120; top: 120; }
  DIV.link4 { position: absolute; left: 65; top: 180; }
```

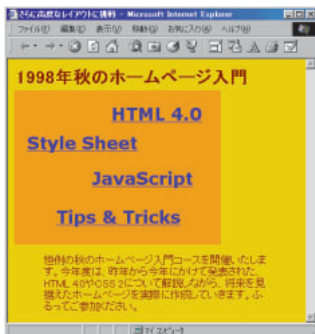
```
2 <DIV CLASS="map">
  <DIV CLASS="link1"><A HREF="1.html">HTML 4.0</A></DIV>
  <DIV CLASS="link2"><A HREF="2.html">Style Sheet</A></DIV>
  <DIV CLASS="link3"><A HREF="3.html">JavaScript</A></DIV>
  <DIV CLASS="link4"><A HREF="4.html">Tips & Tricks</A></DIV>
</DIV>
```

次に、子になる<DIV>タグには「link1」などのクラス名を付けてスタイルを作る。子の場合は、「position: absolute;」とし、「left」プロパティと「top」プロパティで位置を指定する。こうしてのスタイルシートができる。

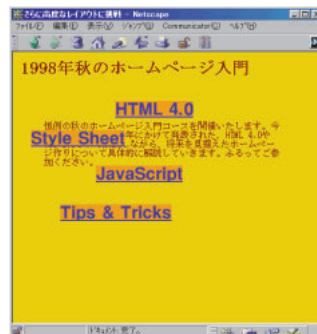
HTMLでは、のように親の<DIV>タグの中に子の<DIV>タグを書けば、親ブロックの内部に子が配置されたデザインができ上がる。子の<DIV>タグ内に画像を置いてもいいが、ここでは簡単にテキストのみとした。

スタイルシートをサポートしていないブラウザでは、<DIV>タグの内容は順に並べられて普通のHTMLになる。イメージマップを作らなくても自由な配置ができ、しかもマップをサポートしないブラウザのためにリンクを別に作ったりしなくてもすむわけだ。残念ながら、ナビゲーター4ではこのサンプルは表示が崩れて正しく表示されなかった。

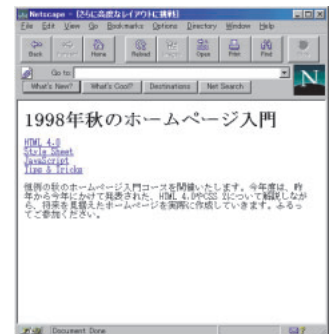
「position」プロパティを組み合わせた自由な配置がナビゲーターでもサポートされるようになれば、ホームページは完全にスタイルシート化できるようになるだろう。



IE4 オレンジ色の親の<DIV>タグの中に子の<DIV>タグが正しく配置される。



ナビゲーター4 「position」プロパティを複雑に組み合わせると表示が乱れる。残念。



ナビゲーター3 スタイルシートをサポートしていないが、正しいHTMLの表示になる。



[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