

HTML MANIA AX

第3回

CSS(スタイルシート)の2つのモード

ウィンドウズ版IE 6、マッキントッシュ版IE 5、そしてネットスケープ6以降には、CSSの表示に2つのモードがある。「標準準拠モード」と「過去互換モード」だ。これは、CSSの標準仕様に厳密に従いながら、古いブラウザとの互換性も保つための機能なのだが、現在ウェブ制作者の頭痛の種となっている。自分のページで最新のブラウザに対応しながらCSSを活用していくには、この2つのモードの違いを十分に研究しておく必要がある。

文:佐藤和人

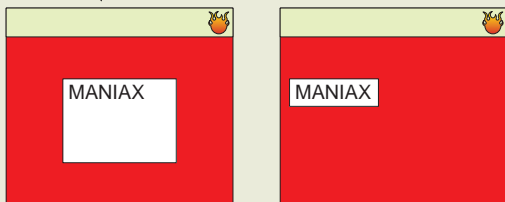
今回紹介したテクニックは internet.impress.co.jp/mani-ax/ でソースを公開!



「標準」と「過去」で天国と地獄!

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
.....
```

標準準拠モード



過去互換モード

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
<head>
.....
```

ソース中のマークは誌面レイアウトの関係で改行していますが、実際はここには改行は入りません。

標準準拠モードは、CSSの仕様書に正確に従った表示をするためのモードだ。過去互換モードは、以前のIEやナビゲーター4での慣習をなぞった表示(仕様に沿っていない間違っただけ)になるモードだ。この2つのモードは、HTML文書の先頭にある文書型宣言(<!DOCTYPE ~>)の種類で切り替わる。どのブラウザでも、左の図のようにシステム識別子("http://~.dtd")のあるHTML 4.01の文書型宣言があると標準準拠モードになる。文書型宣言のないHTMLやHTML 2.0、3.2、またシステム識別子のないHTML 4.01 Transitionalの文書型宣言では過去互換モードになる。

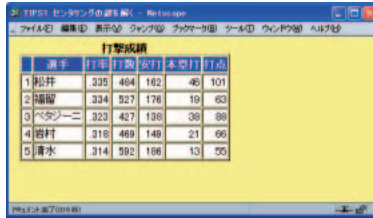
この2つのモードの問題点は、「モードの違いによって表示がどう変わるか」ということ自体がブラウザによって大きく違うことだ。この点がウェブ制作者を混乱させる原因になっている。

[センタリングの謎を解く]

center要素などを使わずに、CSSだけで表をセンタリングするのは、実は難しい。CSSの仕様とブラウザの実際の解釈、標準モードと過去互換モードの違いを把握していないと頭を悩ませることになる。



text-alignでセンタリングしてみるが.....



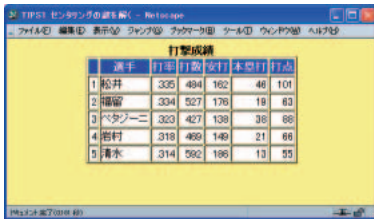
ネットスケープ6、7には効かない。

tips1-1.html (CSS部分)

```
div.block1 { text-align: center; }
th { background-color: #4040FF; color: white; }
td { text-align: right; background-color: white; }
td.player { text-align: left; }
caption { font-weight: bold; }
```

tips1-1.html (HTML部分)

```
<div class="block1"><table>.....</table></div>
```



tableのmargin-leftとmargin-rightをautoにする。



captionのmargin-leftとmargin-rightもautoにする。

tips1-3.html (CSS部分)

```
table, caption { margin-left: auto; margin-right: auto; }
```

ブラウザによる動作の違い

	ブラウザ	標準モード	過去互換モード
margin-leftとmargin-rightにautoを指定	WinIE 6	センタリング	センタリングしない
	MacIE 5	センタリング	センタリング
	ネットスケープ6以降	センタリング	センタリング
text-align: centerを指定したdiv要素でブロック要素を囲む	WinIE 6	センタリング	センタリング
	MacIE 5	センタリングしない	センタリング
	ネットスケープ6以降	センタリングしない	センタリングしない

🔥 text-alignは間違いだ

CSSを使って表をセンタリングしてみよう。ソースのようにdiv要素のtext-alignプロパティをcenterとする。ソースのようにそのdiv要素でtable要素を囲む。するとIEではうまくセンタリングされる。ところがネットスケープ6や7では、表が左に寄ったままだ。これはどうしたことだろう。

実は、text-alignで表がセンタリングされるのは、CSSの解釈としては間違いなのだ。text-alignは「ブロック要素の中身のテキストの揃え方」であって、tableなどのブロック要素をまるごと中央に揃えるものではない。

IEがtext-alignで表をセンタリングするのは、IE独自の解釈にすぎない。ネットスケープ6や7はCSSの仕様に従っているだけなのだ。

🔥 左右の余白をautoにせよ

CSSで表を正しくセンタリングするには、ソースのようにtable要素のmargin-leftとmargin-rightの両プロパティをautoにする。すると、CSSでは左右の余白の大きさは同じと見なされる。表は決まった幅を持っているので、左右に同じ長さの余白が空き、センタリングされることになる。

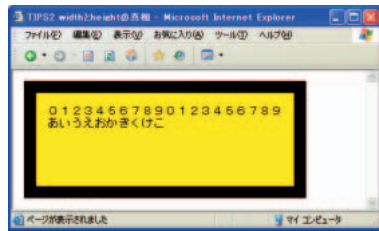
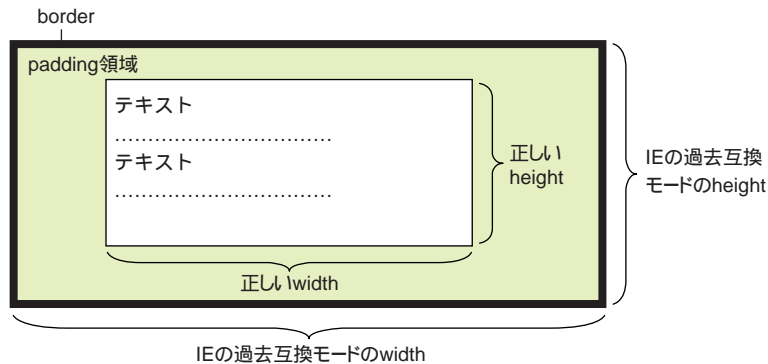
IE 6では、標準モードのときに左右の余白をautoにすると表がセンタリングされる。困ったことに、ブラウザとモードによってどんな動作になるかは、左下の表のようにばらばらだ。古いブラウザでもセンタリングさせたいなら、ソースのようにtext-alignを指定したdiv要素でtable要素を囲んでおく必要がある。

なお、ネットスケープ6と7では、ソースのようにcaption要素でもmargin-leftとmargin-rightをautoにしないと、表のキャプションがセンタリングされないことに注意。

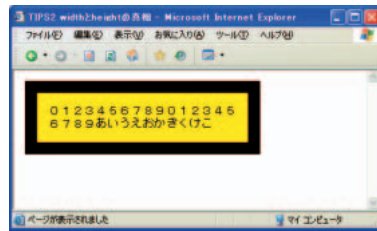
[width と height の真相]

以前にCSSを使ってデザインしたページを新しいブラウザで表示させたときに、表示が大きすぎてしまったときは、2つのモードの違いを疑うといい。幅と高さに関するプロパティに関しては特にそうだ。

width と height の意味



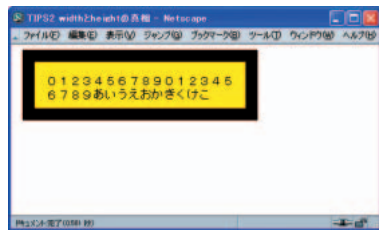
標準準拠モード



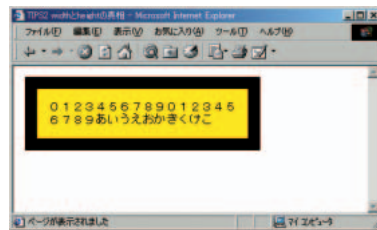
過去互換モード

tips2-1.html、tips2-2.html (CSS部分)

```
#p1 { font-family: monospace;
width: 20em; height: 6em; background-color: yellow;
padding: 1em; border: 1em solid black; }
```



-moz-box-sizingを指定した Netscape 7。



IE 5.5 でもだいたい同じ大きさになる。

tips2-3.html (CSS部分)

```
#p1 { -moz-box-sizing: border-box; }
```

🔥 モードによって サイズが変わる

CSSでwidthやheightプロパティを使うと、要素の幅や高さを固定できる。この「幅」や「高さ」とは、いったいどの部分の幅や高さのことなのだろうか。

CSSの仕様では明快だ。widthとheightは、左の図のように要素の中でテキストを表示する部分の幅と高さだ。ところが古いバージョンのIEは、border(枠線)とpadding(枠線とテキストの間の余白)を含む部分の幅と高さとして解釈してきた。ウィンドウズ版IE 6とマッキントッシュ版IE 5は、標準準拠モードでは仕様に従ってwidthとheightを解釈する。過去互換モードでは、borderやpaddingを含めた幅と高さにする。

ソースでは、フォントを等幅フォントにして2つのモードで表示どう変わるかを実験してみた。widthプロパティに20emを指定しているため、標準準拠モードでは20文字分の全角文字が横におさまる。過去互換モードではborderとpaddingの分だけ幅が狭くなり、16文字しか入らない。

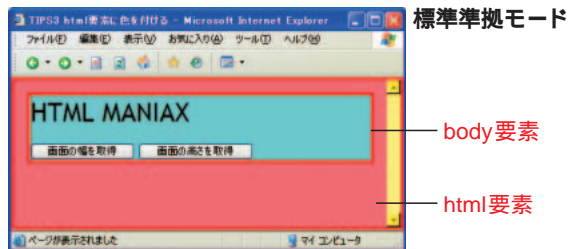
🔥 表示を 統一するには

ネットスケープ6と7は、標準準拠モードでも過去互換モードでもwidthとheightの解釈はCSSの仕様に従う。なので、古いIEとネットスケープ6、7で同じ幅と高さにしようとするのが困ってしまうことになる。

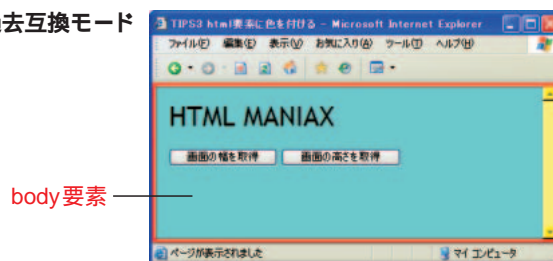
これを解決するには、まず文書型宣言を使ってCSSを過去互換モードにする。これで、古いIEと最新のIEの表示が同じになる。次に、ソースのようにネットスケープ独自のプロパティである-moz-box-sizingを指定して、値をborder-boxとする。するとネットスケープ6、7はwidthやheightを要素の枠線を含めた幅や高さとして解釈する。古いIEと同じような表示にするにはこの裏技を使うしかない。

[色が付くのはhtml? body?]

CSSでスクロールバーに色を付ける機能は、ウィンドウズ版IE 5.5以降の独自機能だが、面白い効果を出せるので人気が高い。しかし、「スクロールバーはどの要素の上にあるのか」を知らないと、混乱することがあるかもしれない。



過去互換モード



tips3-1.html、tips3-2.html (CSS部分)

```
html { background-color: #FF8080;
border: 4px solid red;
scrollbar-base-color: yellow; }
body { margin-left: 1em; margin-right: 1em;
color: black; background-color: #80E0E0;
border: 4px solid red; }
```

IE 5.5、6 どちらでもスクロールバーに色を付ける

```
html, body { scrollbar-base-color: yellow; }
```

tips3-1.html、tips3-2.html (JavaScript)

```
function winWidth() {
  if(window.innerWidth)
    return window.innerWidth;
  else if(document.compatMode == "CSS1Compat")
    return document.documentElement.clientWidth;
  else
    return document.body.clientWidth;
}
```



htmlとbodyに別の背景を付ける

ソース のようにhtmlとbodyに別の背景色を指定したページを見てみよう。CSSの仕様では、ページの内容は「ルート要素」つまりhtml要素の上に乗っていると見なされる。ウィンドウズ版IE 6の標準モードではhtmlとbodyに別々の背景色が付く。過去互換モードではbodyの背景色が全体に塗られる。マッキントッシュ版IE 5とネットスケープ6、7では、2つのモードにかかわらずCSSの仕様に従って別々の背景色が付く。

このようにIE 6では、ページの内容が載っている要素は、標準モードではhtml要素、過去互換モードではbody要素になる。標準モードでは、スクロールバーの色はbodyではなくhtmlに指定すればいい。しかし、htmlに指定するとIE 5.5で反映されなくなってしまう。ソース のようにhtmlとbodyの両方に指定すれば、IE 5.5でもIE 6でも反映される。



JavaScriptでも要注意!

CSSの2つのモードは、JavaScriptを書くときにも影響してくる。ブラウザ画面の幅と高さ取得するには、これまでのIEではdocument.body(つまりbody要素)のclientWidthとclientHeightを参照すればよかった。しかし、IE 6の標準モードでは、document.documentElement(つまりhtml要素)のclientWidthとclientHeightを参照しなければならない。

ソース は、ブラウザの画面の幅を取り出す関数だ。document.compatModeはCSSのモードを調べるプロパティで、値が「CSS1Compat」なら標準モードになる。ネットスケープ、標準モードのIE 6、それ以外のIEで共通して画面の幅を調べるには、こうした関数を用意するといいい。



[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