2014.09.05　冨松

**marginとpadding**

**＜ボックスモデル＞**

CSSではhtmlの全ての要素（タグ）は、右図のような「**ボックス**」と呼ばれる四角い領域を持つと考え、この領域に対して大きさや色、位置の指定をすることでスタイルを指定します。

このボックスは 内容（**content**）、ボーダー（**border**）、パディング（**padding**）、マージン（**margin**）から成ります。contentは文字や画像、borderはそれを囲む枠線、paddingとmarginは余白です。borderは全ての要素に存在するとう考え方がユニークです。

上図の四角部分と、**背景**（背景色・背景画像）との関係を下図に示します。背景画像が前面に、背景色が背面に表示されることに注目してください。

height

width

**＜marginプロパティとpaddingプロパティ＞**

背景に注目して上図を見てください。**padding**はcontent（テキスト）とborderとの余白（**背景部分**）であり、**margin**はborderと隣の要素との余白（**背景なし**）です。これらの余白を指定するプロパティが、それぞれpaddingプロパティとmarginプロパティです。初期値は共に「0」です。このプロパティは全ての要素（タグ）に適用できますが、tableを除く表関連の要素（tr・tdなど）には適用できません。

下図はaaとbbの2つの文字列が、初期化されたp要素 （ブラウザのデフォルトスタイルの項を参照）で指定されているとします。このp要素は青色の背景（下図は背景の外側に幅0のborderがあると考えます）を設定しており、文字列の下側に余白をつける場合ですが、2つのプロパティの違いを確認できます。



下図は南支部だよりで画像を文書の右に配置し、かつ画像の右に10pxのmarginを付けた例です。

<img src="image/donguri.gif" width="100" height="56" style="float:right; margin-right: 10px">

右図の背景c.gifは親要素の背景であり、img要素は背景を設定していません。背景のない要素では、どちらのプロパティを使って余白を作っても見た目は変わりません。ただ、一般の要素は背景なしで使うことが多く、余白の指定はmarginをお勧めします。

なお、画像の配置や本格的なCSSレイアウトに使用する「float」プロパティについては、稿を改めて学ぶことにします。

**＜marginの相殺＞**

paddingにない機能として[marginの相殺](http://kojika17.com/2012/08/margin-of-css.html) があります。下図もaaとbbの2つの文字列が、初期化されたp要素で指定されているとします。この場合paddingでは要素間が10px+10px=20pxとなります。一方、marginでは互いに相殺によって要素間が10pxしか空きません。

このように兄弟要素間の上下方向のmarginは、互いに相殺されます。そのときのmarginは値の大きい方が適用されます。なお、水平方向のmarginの相殺は行われませんので注意してください。



**＜marginの中央配置＞**

横幅を指定した状態でmargin-left, margin-right にautoを指定した場合、左右等分のmarginが算出されボックスは中央に配置されます。（marginに対してautoをすると値は0になります）

右図で3つの記述は、いずれを使っても中央揃えになります。ただ、上段の記述は少し長くなりますし、中段の記述例は何故か少ないです。

margin-left: auto; margin-right: auto;

margin: auto;

margin: 0 auto;

一般には、下段の「margin: 0 auto;」と記述した例が多いようです。

**＜ブラウザのデフォルトスタイルシート＞**

一般のブラウザでは、あらかじめスタイルを定義した「[デフォルトスタイルシート](http://kojika17.com/2012/09/reset-css-from-default-style-sheet.html)」を持っています。右図にその一例を示します。

body {

 margin: 8px;

}

h1 {

 margin: 0.67em 0;

 font-size: 2em;

 font-weight: bold;

}

h2 {

 margin: 0.83em 0;

 font-size: 1.5em;

 font-weight: bold;

}

h3 {

 margin: 1em 0;

 font-size: 1.17em;

 font-weight: bold;

}

p {

 margin: 1em 0;

}

ul {/\* olもほぼ同じ \*/

 margin: 1em 0;

 padding-left: 30px;

 list-style-type: disc;

}

strong {

 font-weight: bold;

}

em {

 font-weight: italic;

}

枚方HPでもよく使う 段落「p」や見出し「h1～h6」では、ブラウザにデフォルトのスタイルがあることに注目してください。

このデフォルトスタイルは、多くのブロックレベル要素に、上下のmarginを指定していますが、paddingの指定は少ないです。

**・指定が重なった場合の優先順位**

CSSによる記述 ＞htmlによる記述 ＞ブラウザのデフォルトスタイルの順になります。

**・marginと paddingの初期化**

hira-m.cssのbodyセレクタには下記の指定があります。

body { margin: 0px; padding: 0px; }

この指定はmarginとpaddingのデフォルト値がブラウザによって違うので、それを統一するための設定です。ただ、注意して欲しいのは、このデフォルトスタイルシートはよく考えられて作られているということです。一般的には初期化せず、そのまま利用すると便利にできています。

**・ulとolにおける初期化**

[ulとol](http://www.tagindex.com/stylesheet/list/margin_padding.html) は上図のように、一般のブラウザではデフォルトスタイルが設定されています。ulでリストマーカー不要の場合はmarginとpaddingを初期化してもOKですが、リストマーカーを付ける場合やolで番号を付ける場合は、これらは親ボックスから左側にはみ出てしまいます。

そこで、左側にマーカーや番号の1つ分ほどのmarginを入れます。デフォルトのmarginとpaddingを0にして、左側にだけ適当なmarginを設定するといった感じになります。

ul（マーカーなし）

ul{

 list-style-type: none;

 margin: 0;

 padding: 0;

}

ul（マーカーあり）

ul{

 margin: 0 0 0 1em;

 padding: 0;

}

ol（番号あり）

ol{

 margin: 0 0 0 2em;

 padding: 0;

}

以上