



第8回 装飾表現は複数のCSSを組み合わせせて

CSSを使った装飾表現は、すべて複数のプロパティの組み合わせによるものです。ここまで紹介してきたボックスモデルの幅や高さ、余白やボーダー、前景色と背景色、テキストの装飾などのプロパティを組み合わせ、さまざまな表現をしていきます。ただ、使用頻度の高い代表的なプロパティの組み合わせだけでは、インパクトに欠ける場合もあるでしょう。今回は文字装飾の場合を例に、ここまで覚えていろいろな用途のプロパティを組み合わせ、オリジナルの表現を作ってみましょう。

(解説：こもりまさあき)

ボーダーと余白を組み合わせで見出しを表現

見出しを表現する場合、前回紹介したフォントやテキストに関するプロパティを利用し、フォントを太字にして文字サイズを大きくするのが通常の表現です。もちろん、それだけではちょっとインパクトが足りないこともあるでしょう。見出しとなるテキストは、ほかのテキストブロックとの差別化が必要なため、colorプロパティを使って文字色を変更する装飾指示を加える場合も多いです。さらに視覚的に違いを出すために、ボーダーや余白を使ってみるのもひとつの方法でしょう。

連載第5回の「ボックスモデル」の項を思い出してください。見出しとなるh要素に限らず、ページ内に配置される多くの要素はブロックレベル要素であり、それぞれ初期表示として目には見えないボーダーや余白、マージンが付いていることを説明しました。これらはテキストなどの内容を囲む四辺に存在しています。見出しにインパクトを付ける方法として、内容を囲むように上下左右のすべてのボーダーを有効にしたり、下だけにボーダーを適用したりすることは常套手段です。インパクトを出すために見出しを画像化することでもできますが、更新や修正が頻繁に発生しがちなものはCSSだけの装飾に留めておくのがコスト削減と作業の効率化に繋がるでしょう。

【図1】

【HTML ソース】

```
(中略)
<body>
<h1> 四方をボーダーで囲んだ見出し </h1>
<h2> 左と下のボーダーで付箋風に装飾した見出し </h2>
</body>
</html>
```

【CSS ソース】

```
(中略)
h1 {
width: 480px;
margin: 0;
padding: 0 0.5em;
color: #333333;
font-size: 1em;
line-height: 2;
border: 1px solid #333333;
}
h2 {
width: 320px;
margin: 0;
padding: 0 0 0 10px;
color: #222222;
font-size: 0.9em;
line-height: 2.2;
border-top: 1px solid #cc0000;
border-right: 1px solid #cc0000;
border-bottom: 1px solid #cc0000;
border-left: 12px solid #cc0000;
}
```



見出しを画像化することが現実的ではない場合、ボーダーや余白を調整したり背景画像を適用したりして装飾することが多い。複数のプロパティをデザインに合わせて適用しよう。ここでは、h1要素のborderの指定に「border: 1px solid #333333;」として、ショートハンドによる記述を行っている。これで四方すべてに同じ線種のボーダーが適用される

【図2】

【HTML ソース】

（中略）

```
<body>
<p class="solid">border-style: solid</p>
<p class="dotted">border-style: dotted</p>
<p class="dashed">border-style: dashed</p>
<p class="groove">border-style: groove</p>
<p class="ridge">border-style: ridge</p>
</body>
```

【CSS ソース】

（中略）

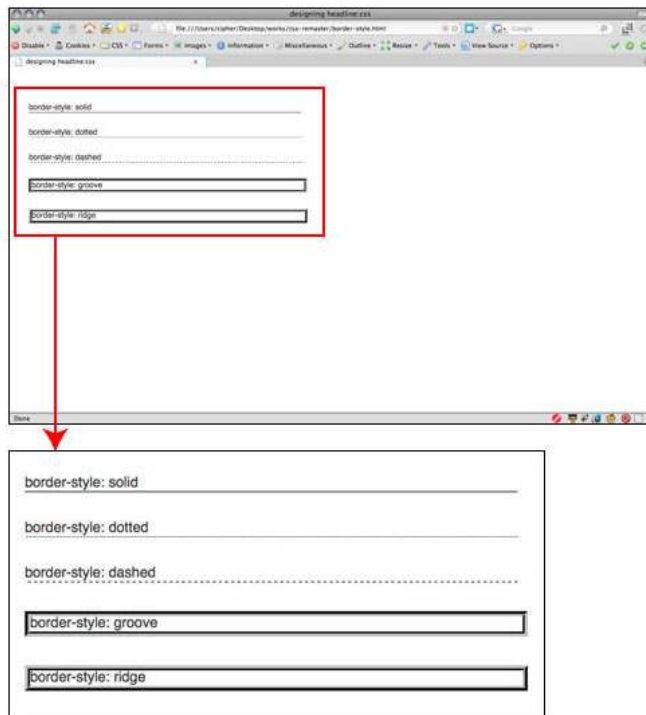
```
p.solid {
border-bottom: 1px solid #333333;
}

p.dotted {
border-bottom: 1px dotted #333333;
}

p.dashed {
border-bottom: 1px dashed #333333;
}

p.groove {
border: 5px groove #333333;
}

p.ridge {
border: 5px ridge #333333;
}
```



ボーダーのスタイルは「solid」、「dotted」、「dashed」、「double」、「groove」、「ridge」、「inset」、「outset」を指定可能だ。「double」は二重線になり「1px」では効果が視認できないので、最低でも線幅に「3px」を指定する

table要素の装飾もプロパティの組み合わせ次第

Webページ内で価格表やサイズ表のような表組みデータをマークアップする場合に使用するtable要素は、CSSを用いることで従来のHTMLの属性指定だけでは不可能だった装飾表現も適用可能になります。HTMLだけでは、表組み全体のボーダーを有効にするかどうかのborder属性、td要素であるセル同士の間隔を調整するcellspacing属性、セル内の余白を調整するcellpadding属性を組み合わせるぐらいしか表現方法はありませんでした。CSSを利用すれば、もっと細やかな表組みの装飾が可能になります。

表組みの装飾に使用するのも、borderやpaddingといったプロパティです。CSSではtable要素だけでなく、それに内包されるtd要素へも上下左右のボーダーや余白を設定することができます。つまり、CSSの適用先とそのプロパティ次第で多様な表現ができるということです。従来のHTMLでの装飾では自由度が低く感じるが多かった表組みも、サイトのデザインテイストに合わせた装飾ができると覚えておきましょう。すべてはプロパティの組み合わせ次第なのです。

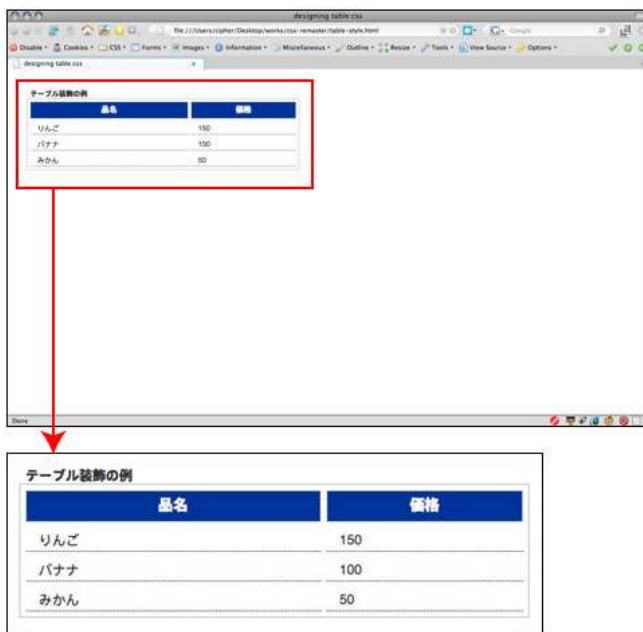
【図3】

【HTML ソース】

```
(中略)
<body>
<table summary=" テーブル装飾の例 ">
<caption> テーブル装飾の例 </caption>
<tr><th> 品名 </th><th> 価格 </th></tr>
<tr><td> りんご </td><td>150</td></tr>
<tr><td> バナナ </td><td>100</td></tr>
<tr><td> みかん </td><td>50</td></tr>
</table>
</body>
</html>
```

【CSS ソース】

```
(中略)
table {
width: 480px;
height: auto;
border-collapse: separate;
border-spacing: 5px;
border: 1px solid #c0c0c0;
}
table caption {
caption-side: top;
padding-left: 6px;
text-align: left;
font-weight: bold;
font-size: 0.9em;
color: #333333;
}
table th {
padding: 0;
border: 1px solid #c0c0c0;
color: #ffffff;
background-color: #003399;
font-weight: bold;
line-height: 2em;
text-align: center;
}
table td {
padding: 5px 0 2px 1em;
border-bottom: 1px dotted #333333;
color: #333333;
font-size: 0.9em;
text-align: left;
}
```



従来の表組みの装飾では、HTMLに用意されていたborder属性、cellpadding属性、cellspacing属性とtd要素に対するbgcolor属性などを使って装飾指示を行うことが多かったはずだ。これらもすべてCSSで置き換えることができる。table要素で、「border-collapse」プロパティの値に「collapse」、もしくは「separate」を指示することで、表組み内のセル同士の間を「閉じる」、「開く」の表現ができる。「border」プロパティを使えば、表の周囲に罫線を表示できる。「border-spacing」プロパティはcellspacing属性の置き換えだと考えてもよいだろう。値を指定すれば、「border-collapse: separate;」の状態の間隔を調整できる。それ以外のth要素やtd要素に対しては、ここまで紹介した多くの装飾指示が指定されていることがわかるはずだ

今回はCSSを使った装飾のまとめとして、いろいろなプロパティを組み合わせるという手法を紹介しました。前回、前々回と紹介したフォントやテキストに関するプロパティ、前景色や背景色、それに余白やボーダーを組み合わせることで多様な表現が可能になります。そして、装飾はこれらのプロパティの組み合わせ次第なのです。いろいろな組み合わせを試して、オリジナルの表現を作り上げましょう。

Point

- ・ 画像化しにくい要素は、ボーダーや余白を駆使して装飾する
- ・ table要素もプロパティの組み合わせで多様な表現ができる
- ・ 要素の装飾はプロパティの組み合わせがすべて

次回からは、いよいよCSSで一番つまずきやすいレイアウトについての解説をスタートします。
