



第9回 レイアウトの基本をマスターしよう

ここからは、CSSで一番躓きやすいレイアウトの解説をはじめます。CSSによる装飾はブラウザの種別などにかかわらず、指示したスタイルがある程度同じように適用されますが、レイアウトとなると話はちょっと異なります。レイアウトの考え方は非常に単純なのですが、ブラウザ種別による表示結果の違いに悩まされることもあります。そこで今回は、CSSレイアウトの基本とも言える「floatプロパティ」を使った「テキストの回り込み」を表現してみましょう。

(解説：こもりまさあき)

考え方を理解すれば、レイアウトは怖くない

HTML+CSSでWebページを制作し始めたばかりの方がよく聞くのは、CSSでのレイアウトにすることがほとんどではないでしょうか。装飾表現に関しては、ブラウザ毎に多少の表示上の違いはあっても、指定したスタイルはほとんど意図したように表示されます。しかし、CSSを使ったレイアウトは色を付けたりといった装飾表現とは別物で、その考え方と仕組みに慣れないうちはなかなか理解しがたいものです。

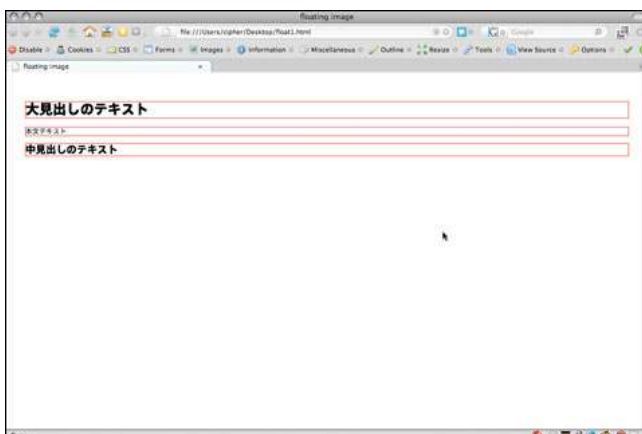
レイアウトに入る前に本連載の「第1回」と「第5回」の話を思い出しましょう。本来HTML文書というものは、ページ内に含まれる内容に対し情報構造上の意味を与えるため、内容に相応しいHTMLタグ（要素）でマークアップされた文書です。情報構造上の意味を持たせるためには、内容の多くをブロックレベル要素でマークアップします。その結果、装飾指示のないHTMLは、ブラウザウィンドウ内の横の表示領域全体を使ってひとつのブロックを形成し、文書上の順番に従って縦方向に並んで表示されます。

CSSを使ったレイアウトとは、この縦に整列したボックスを「意図した場所に、意図した大きさと配置する」だけなのです。レイアウトに使用するプロパティの数は多くはありませんし、手法についても順を追って理解すれば決して難しいものではありません。さっそく、CSSレイアウトの基本中の基本ともいえる「テキストの回り込み」の表現にチャレンジしてみましょう。

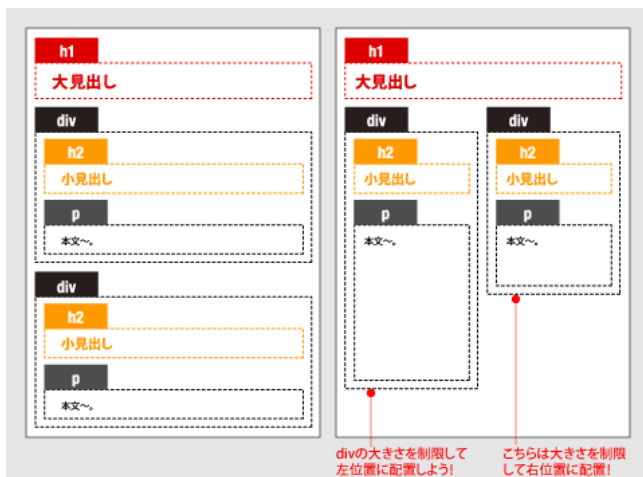
【HTML ソース】

(中略)

```
<body>
<h1> 大見出しのテキスト </h1>
<p> 本文テキスト </p>
<h2> 中見出しのテキスト </h2>
</body>
</html>
```



HTML文書中にブロックレベル要素としてマークアップされた内容は、ブラウザの横幅一杯に広がるボックスとして要素を配置し、文書中の記述順で縦に整列する



CSSを使ったレイアウトは、HTMLに記述された情報構造をもとに個々の要素の大きさを調整しながら、意図した位置に表示するのが基本。テーブルレイアウトのように先にブロックの配置位置を決定してからマークアップすることもできるが、HTMLとCSSについて十分な理解がないと完成した文書の情報構造がおかしくなる可能性も出てくるため、あまりオススメできない

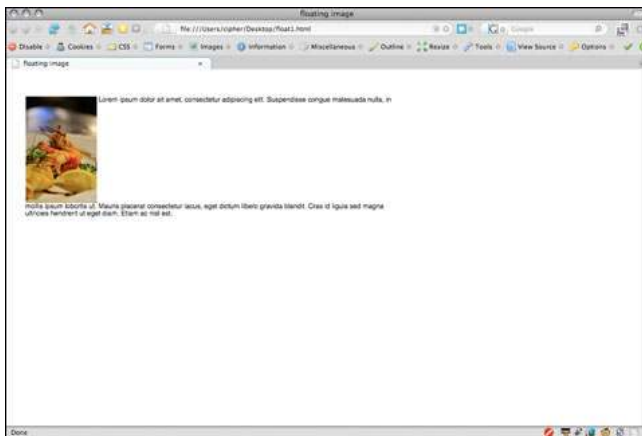
「float」プロパティを使ったテキストの回り込み

さっそく、CSSレイアウトの基本中の基本ともいえる「テキストの回り込み」を表現しましょう。例えば、ある画像の周囲に複数行にわたるテキストを回り込ませたい場合、従来HTML側に指示していたalign属性を使ってうまく表示できません。これは画像の後にテキストが続いたとしても、インライン要素が横1行に並んでいるだけに過ぎません。align属性は位置関係を指定するだけですから、複数行を回り込ませるような表現にはならないのです。

【HTML ソース】

(中略)

```
<body>
<p> 本文テキスト </p>
</body>
</html>
```



HTML中にalign属性を指定しても、複数行にわたるテキストを画像の横に表示することはできない。align属性は視覚的な位置を指定するものであるため、HTMLには文書構造だけを記述すると考えれば指定しない方がよい

画像の横に複数行のテキストを回り込ませる場合は、CSSの「float」プロパティを使います。floatプロパティは、「left」、「right」の値を指定することで、それが含まれるボックスを左右へ移動させることができます。widthプロパティとheightプロパティを使えば、大きさを制限したボックスをコントロール可能です。「フロート」といえば、コーヒーフロートのアイスクリームのようにプカプカ浮いているものだと考えると理解しやすいかもしれません。floatプロパティを指定した要素は、プカプカと浮いたような状態になって後に続く内容が上にスライドして表示されるのです。

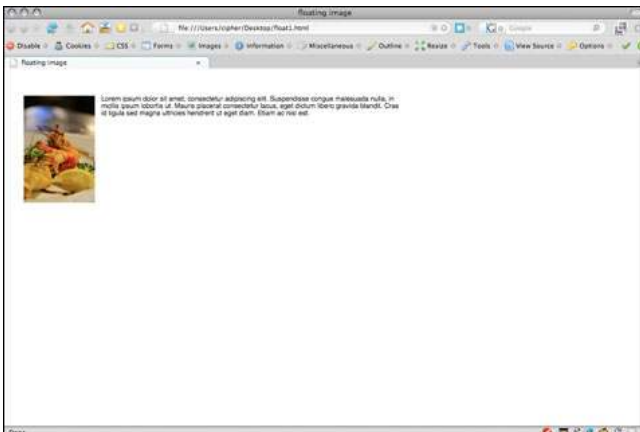
【HTML ソース】

(中略)

```
<body>
<p class="photos"></p>
<p> 本文テキスト </p>
</body>
</html>
```

【CSS ソース】

```
p img {
float: left;
width: 120px;
height: 180px;
margin: 0 1em 1em 0;
}
```



img要素に直接floatプロパティを指定しても、テキストの回り込みは表現可能。この場合、HTML中のimg要素に確実に幅と高さを明示するか、CSS側でも指定した画像と同じ大きさのwidthとheightの値を明示しておきたい

【HTML ソース】

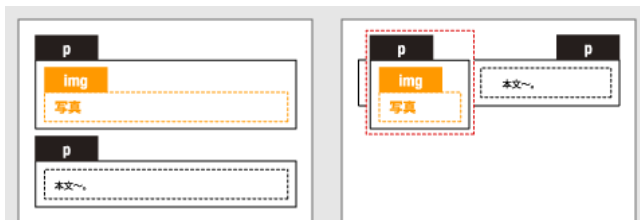
(中略)

```
<h2> 中見出しのテキスト </h2>
<p class="photos"></p>
<p> 本文テキスト </p>
<h3> 小見出しのテキスト </h3>
</body>
</html>
```

【CSS ソース】

```
p.photos {
float: left;
width: auto;
height: auto;
margin: 0 1em 1em 0;
}
```

情報構造上、それぞれを別のブロックレベル要素でマークアップしている場合は、class属性などを使って指定しよう。この場合はHTMLのp要素内のimg要素に幅と高さが指定されていれば、CSS側はp要素の内容の幅で自動調整されるので「auto」でも構わない



ブロックレベル要素の大きさとfloatプロパティでフロートさせる方向を指示すれば、そのボックスは任意の大きさと固定され浮いたような状態になる。本来ボックスが伸びていた部分には空きができるため、その位置に後に続く要素がスライドして表示されることになる。図のように写真の入るブロックのみ大きさを制限した場合は、後に続く要素がこのように重なっているイメージになる

このようにfloatプロパティを使えば、テキストの回り込みが簡単に実現可能です。「ボックスの幅を制限してフロートさせることで、後に続く内容が上にスライドする」と覚えておきましょう。

フロートを解除するのは「clear」プロパティ

floatプロパティは非常に便利なプロパティで、フロートを指定した要素の後に続く要素を空きスペースに流し込むことが可能です。しかし、本当は横に配置したくない要素までも流れてしまっては問題です。例えば、下記のサンプルのように小見出しにあたるh3要素が画像の横に配置されているのは問題になることもあります。デザインラフでは画像の下に配置するべきなのに、フロートを指定したことによって前後の要素に含まれる内容量の関係でこのような結果になることもあります。

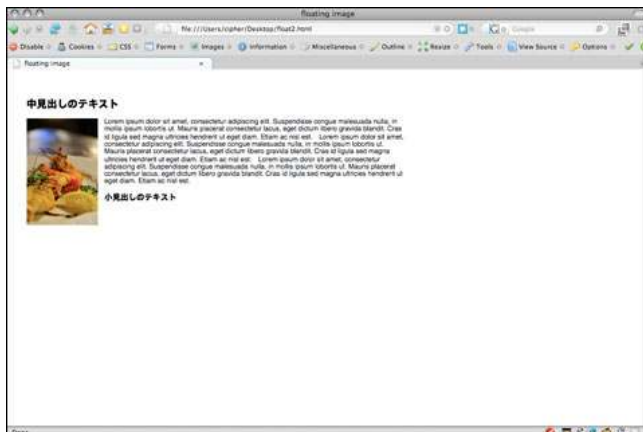
【HTML ソース】

(中略)

```
<h2> 中見出しのテキスト </h2>
<p class="photos"></p>
<p> 本文テキスト </p>
<h3> 小見出しのテキスト </h3>
</body>
</html>
```

【CSS ソース】

```
p.photos {
float: left;
width: auto;
height: auto;
margin: 0 1em 1em 0;
}
```



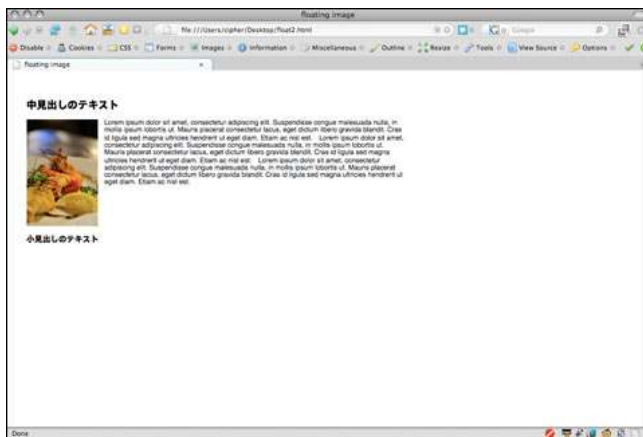
一旦フロートを指定すると、後に続く要素はそれを解除しない限り、空いているスペースに流れてしまう。デザインラフで小見出しのテキストが本文の下に配置されるべきなら、このサンプルのように本文の後に続いて空きスペースに小見出しが表示されては問題だ

このようにフロートさせることによってできた空きスペースに特定の要素を流し込みたくない場合は、そのブロックレベル要素に対して「clear」プロパティを指定します。clearプロパティは、floatプロパティで指示されたフロート指定を解除します。値には、float指定した「left」「right」のいずれかか、双方を解除する「both」を指定しましょう。

【CSS ソース】

```
p.photos {
float: left;
width: auto;
height: auto;
margin: 0 1em 1em 0;
}

h3 {
clear: left;
}
```



写真と本文のブロックの下に小見出しを配置するためには、小見出しでフロートを解除する必要がある。ここではh3に対して「clear: left;」を指定した

内容によって必ずしもクリアが必要な場合ばかりではありませんが、このサンプルのように簡単なレイアウトであることは稀です。「フロートさせたらどこかでクリアする」ということも併せて覚えておきましょう。フロートを解除することを忘れていることによって、意図したレイアウトにならないということは起きやすいので注意が必要でしょう。

Point

- ・CSSでのレイアウトとは「マークアップされたブロックをどう並べるか」と考える
- ・floatプロパティは、特定のブロックを浮かせて左右に移動させる
- ・フロートさせたくないブロックが出てきたら、clearプロパティを指定する

今回はレイアウトの基本ということで、floatプロパティを使った「テキストの回り込み」の表現とclearプロパティについて解説しました。いよいよ次回からは、本格的なページレイアウト手法の解説に入ります。楽しみに。