

情報量の多いニュースサイトや 企業サイトで見られるfloatを使った 3カラムレイアウトの指定方法

近年では、表示解像度の高いモニタ^{※1}が普及しています。結果ウェブページも広い表示領域を使う3カラム型のレイアウトが受け入れられるようになりました。情報量の多いニュース系サイトやブログサイトはもちろん、企業サイトでも採用するケースが増えてきています。作り方の基本的な概念は2カラムレイアウトと変わりませんが、XHTMLに工夫するとより柔軟性のあるページが作れます。

3カラムレイアウトの 基礎知識

3カラムレイアウトには3種類の構成パターンが考えられます。1つは中央にコンテンツ領域を持ちサイドバーが左右に存在するタイプ、2つめはサイドバーが左側に2つ並んでいるタイプ、サイドバーが右側に2つ並んでいるタイプの3つです(01)。

02のXHTMLを見てください。この3カラムレイアウトで実装することを前提としたXHTMLの特徴は、CSSを書き換えるだけで3種類とも実現できる汎用的な構造を採用した点です。2カラムレイアウトのXHTML(「056 floatを使った基本の2カラムレイアウト」参照)と比較すると、divが2つ追加されています。1つは、div#mainとdiv#subを内包するdiv#contents、もう1つは、div#contentsのあとに出現するdiv#extraです。

まず各大枠となるdiv要素の幅を指定しましょう(03)。とくに親要素と子要素の幅設定に矛盾が生じないように設計します。下準備が不十分では余白が大きく付き過ぎたり、あるカラムが下に落ちてしまう「カラム落ち」^{※2}が発生するので注意が必要です。現時点ではまだfloat処理を施していないのでカラムが縦に並んでいます。

02

body要素以下のXHTML

```
<body>
<div id="wrapper">
  <div id="header"></div>
  <div id="container">
    <div id="contents">
      <div id="main"></div>
      <div id="sub"></div>
    </div>
    <div id="extra"></div>
  </div>
  <div id="footer"></div>
</div></body>
```

01

3カラムレイアウトの種類



floatの指定をして 3カラムレイアウトを完成させよう

では親のdiv要素であるdiv#contentsとdiv#extraに注目しましょう。最初にこれらのdivで2カラムレイアウトを作成します。div#contentsに対して「float:left;」を、div#extraに対して「float:right;」を指定します(04)。

次に、div#contentsが内包するdiv#mainとdiv#subにフロートを指定します。div#mainに対して「float:right;」を、div#subに対して「float:left;」を指定します。親と子両方のdivで2カラムレイアウトを作るわけですが、親は2つのカラムを1つのカラムの横にfloatさせる役割だったことがわかるでしょう。仕上げに、div#containerに対して、「clearfix」の指定を忘れずに行います(05)。

なお、サイドバーが右側に2つ並んでいる3カラムレイアウトでの場合は、05で作成したサンプルのdiv#mainを「float:left;」、div#subを「float:right;」に変更します。また、サイドバーが左側に2つ並んでいるレイアウトの場合は、div#contentsに「float:right;」、div#extraに「float:left;」を指定してdiv#mainを一番右側へ配置します。

※1 モニタサイズ

モニタの一般的なサイズには、SVGA(800×600)、XGA(1024×768)、SXGA(1280×1024)などがあります。長年SVGA(800×600)への対応が求められていましたが、近年ではユーザーのモニタサイズの拡大に伴い、最低ラインをXGA(1024×768)に設定する案件がほとんどです。

※2 カラム落ちが発生する条件

カラム落ちが発生してしまう原因として主に考えられるのは、子要素の横幅の合計が親要素より大きくなってしまった場合です。その他にも、著しく長い半角英数字の文字列がある場合や、列数の多い表組み(table)がある場合など、原因はさまざまです(詳しくは「061 カラム落ち発生時の対処方法」参照)。

03

CSS

```
div#wrapper {
width: 840px;
}
div#contents {
width: 636px;
}
div#main {
width: 432px;
}
div#sub {
width: 192px;
}
div#extra {
width: 192px;
}
```

各divに幅を指定する

04

CSS

```
div#wrapper {
width: 840px;
}
div#contents {
float: left; width: 636px;
}
div#main {
width: 432px;
}
div#sub {
width: 192px;
}
div#extra {
float: right;
width: 192px;
}
```

div#contentsとdiv#extraを使って
2カラムにした状態

05

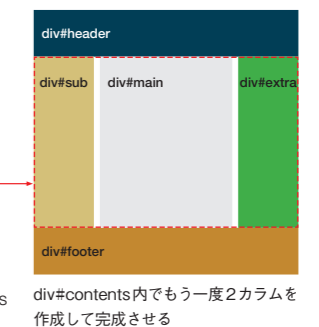
CSS

```
div#wrapper {
width: 840px;
}
div#contents {
float: left; width: 636px;
}
div#main {
float: right;
width: 432px;
}
```

```
div#sub {
float: left;
width: 192px;
}
div#extra {
float: right;
width: 192px;
}
```

```
div#container:after {
display: block;
clear: both;
height: 0;
visibility: hidden;
content: ".";
}
```

div#contents

div#contents内でもう一度2カラムを
作成して完成させる