

# CSS Multi-column Layout Module Level 1

W3C Working Draft, 12 February 2021

## 抄録

本仕様書は、Web 用スタイルシート言語である CSS における複数カラムレイアウトについて規定する。本仕様で規定されている機能を利用して、コンテンツを複数の列の間に隙間やルールを設けて流すことができる。

CSSは、構造化された文書（HTMLやXMLなど）の画面や紙面などでの描画を記述するための言語である。

## この文書のステータス

このセクションでは、この文書の発行時の状態を説明します。他の文書がこの文書に取って代わることがある。現在のW3C出版物及びこの技術報告書の最新改訂版のリストは、<https://www.w3.org/TR/> のW3C技術報告書インデックスに掲載されている。

この文書は、CSS作業部会が作業草案として発行したものである。ワーキングドラフトとして出版されたことは、W3C メンバーシップによる承認を意味するものではない。

この文書は草案であり、いつでも他の文書に更新されたり、置き換えられたり、廃止されたりする可能性がある。この文書を作業中以外のものとして引用することは不適切です。

以下のように、タイトルに仕様コード "css-multicol" を含めて GitHub に課題を提出することでフィードバックを送ってください (推奨)。"[css-multicol] ...コメントの要約..." のようにしてください。すべての問題とコメントはアーカイブされます。また、(アーカイブされた)公開メーリングリスト [www-style@w3.org](mailto:www-style@w3.org) にフィードバックを送ることもできます。

この文書は、2020年9月15日のW3Cプロセス文書に準拠する。

この文書は、W3C特許ポリシーの下で運営されているグループによって作成された。W3Cは、グループの成果物に関連して行われた特許開示の公開リストを維持している。本質的クレームを含むと考えられる特許を実際に知っている個人は、W3C特許ポリシーの第6項に従って情報を開示しなければならない。

## 1. 序章

(このセクションは規範ではありません)。

このモジュールでは、CSSにおける複数列レイアウトについて説明する。このドキュメントで説明する機能を使用することで、スタイルシートは、要素の内容を複数の列にレイアウトすることを宣言することができます。

CSS の他のレイアウト方法は、親要素に適用されると、直接の子要素の表示プロパティを変更する。例えば、3列のグリッド・レイアウトが作成された場合、グリッド・コンテナの直接の子要素はグリッド・アイテムとなり、必要に応じて追加の行が作成され、セルごとに 1つの要素がカラム・トラックに配置されます。

しかし、マルチカラムコンテナの子要素は通常のフローを継続し、そのフローは複数のカラムに配置されます。これらの列は柔軟なインラインサイズを持っているため、表示される列のサイズや数を変更することで利用可能なスペースに対応します。

複数カラムのレイアウトは、CSSで記述するのは簡単です。以下に簡単な例を示します。

### 例1

```
body { column-width: 12em }
```

正確なカラム数は利用可能なスペースに依存します。

カラム数はスタイルシートで明示的に設定することもできます。

### 例2

```
body { column-count: 2 }
```

この場合、カラムの数は固定で、カラムの幅は利用可能な幅に応じて変化します。

短縮型のcolumnsプロパティを使用して、1つの宣言でどちらか、または両方のプロパティを設定することができます。

### 例3

これらの例では、カラムの数、カラムの幅、および数と幅の両方がそれぞれ設定されています。

```
body { columns: 2 }  
body { columns: 12em }  
body { columns: 2 12em }
```

このモジュールで導入されたもう一つのプロパティ群は、カラム間のギャップとルールを記述します。

### 例4

```
body {  
  column-gap: 1em;  
  column-rule: thin solid black;  
}
```

上の例の最初の宣言では、隣接する2つの列の間のギャップを1emに設定しています。列の隙間はバディング領域に似ています。隙間の中央には、column-ruleプロパティによって記述されるルールがあります。

column-ruleプロパティの値は、CSSのborderプロパティの値と似ています。borderと同様に、column-rule は短縮プロパティです。

### 例5

この例では、上記の例の短縮型のcolumn-rule宣言を展開しています。

```
body {
  column-gap: 1em;
  column-rule-width: thin;
  column-rule-style: solid;
  column-rule-color: black;
}
```

column-fillとcolumn-spanプロパティは、スタイルシートに複数列レイアウトの視覚的表現の幅を与えます。

### 例6

この例では、列はバランスがとれている、つまりほぼ同じ長さになるように設定されています。また、h2 要素はすべての列にまたがるように設定されています。

```
div { column-fill: balance }
h2 { column-span: all }
```

この仕様では、10個の新しいプロパティを導入していますが、そのすべてが上記の例で使用されています。

すべての列プロパティが初期値を持つ場合、要素のレイアウトは、1つの列だけを持つ複数列レイアウトと同じになります。

### 例7

列の段間（斜めのハッチング）と列の規則は、パディング（十字のハッチング）を持つ複数列コンテナのこのサンプルのレンダリングで示されています。ハッチングされた部分は、説明のためだけに存在しています。実際の実装では、これらの領域は背景によって決定されます。2 番目の画像は、列規則を持つ複数列コンテナのレンダリングを示しています。