

CSS グリッド

CSS3 で新たに追加されたグリッドの使い方を学ぶ

グリッドとは

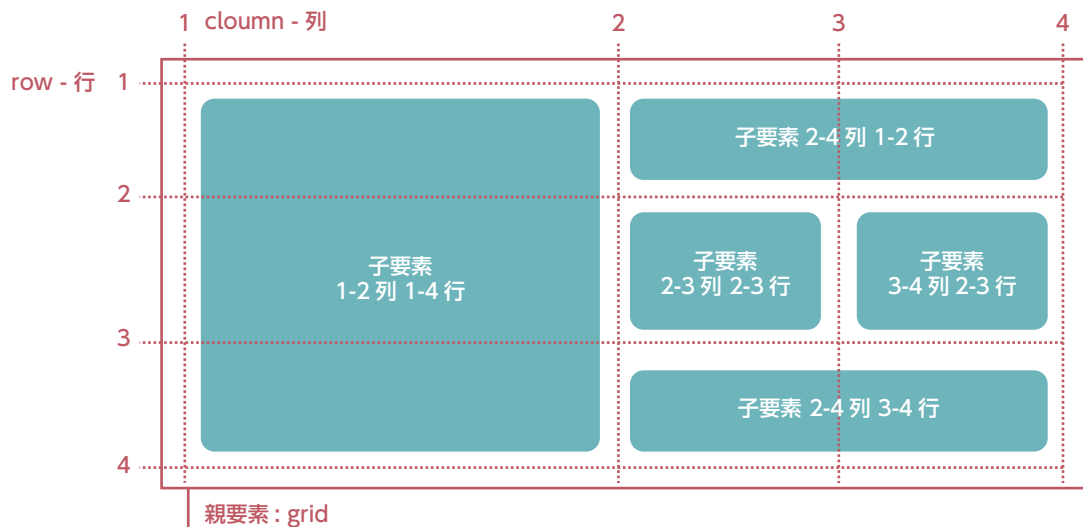
CSS3 から新たに追加された、レイアウトに使用可能な方法です。親要素に指定することで、ページに分割領域を指定し、子要素が分割領域に従って要素の各配置を行います。

```
E { display: grid; }
```

親要素に対して display プロパティで指定する。

grid レイアウトのイメージ

親要素へグリッドを指定し、分割領域を指定。子要素はそれに合わせてレイアウト配置を行う。例に下記図の様なレイアウトを組む場合、4x4の分割領域を作りレイアウトを作成していく。



分割領域の作成

親要素へグリッドを指定し、分割領域を指定。子要素はそれに合わせてレイアウト配置を行う。例に下記図の様なレイアウトを組む場合、4x4の分割領域を作りレイアウトを作成していく。

grid-template-columns / grid-template-rows

親要素に対して、グリッド列（行）の領域指定を行う

値	意味
none（初期値）	負でない長さ値 fr minmax() fit-content repeat() 等指定可能

MDN - grid-template-columns

<https://developer.mozilla.org/ja/docs/Web/CSS/grid-template-columns>

grid-columns / grid-rows

子要素に対してグリッド列（行）の分割指定を行う

値	意味
行（列）番号 / 行（列）番号 初期値：auto	/ は n 番目～n 番目 のから という意味で使⤵します。

4x4の分割数でGridレイアウトの記述

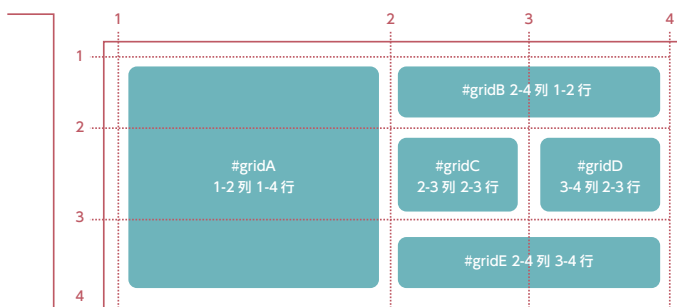
では、実際に先程のレイアウト例を元にコードで記述してみます。サンプルファイルを元に4列4行の分割領域を指定して、レイアウトを作成してみましょう。

sample7_grid.html - HTML

```
<div class="wrapGrid">
  <section id="gridA">
    <h3>page title</h3>
    <p>Lorem ipsum do ...
  ~
```

sample7_grid - CSS

```
.wrapGrid{
  margin: 20px;
  display: grid;
  grid-template-columns: 300px 300px 300px;
  grid-template-rows: 300px 300px 300px;
}
#gridA{
  grid-column: 1 / 2;
  grid-row: 1 / 4;
}
#gridB{
  grid-column: 2 / 4;
  grid-row: 1 / 2;
}
#gridC{
  grid-column: 2 / 3;
  grid-row: 2 / 3;
}
#gridD{
  grid-column: 3 / 4;
  grid-row: 2 / 3;
}
#gridE{
  grid-column: 2 / 4;
  grid-row: 3 / 4;
}
```



先程の表の列・行番号を元に子要素の分割数を指定する。今回はの分割領域は300px x 300px の固定値で表示される。

Gridレイアウト レスポンシブに対応した伸縮レイアウト

px指定した分割数の場合、固定レイアウトになってしまうため、解像度が異なるレイアウトの場合、そのままの表示となってしまいます。そのため、通常のグリッドレイアウトに伸縮出来る設定を追加し、柔軟に動くレイアウトになる様変化をつけましょう。

repeat() 関数

() 内の繰り返しを行う

値	意味
正の整数で表せる長さ	2であれば、2回繰り返す
auto-fit	グリッドコンテナが最大幅を持つ場合、フローしない最大数で埋められる
auto-fill	グリッドコンテナが最大幅を持つ場合、空のコンテナを作り繰り返す

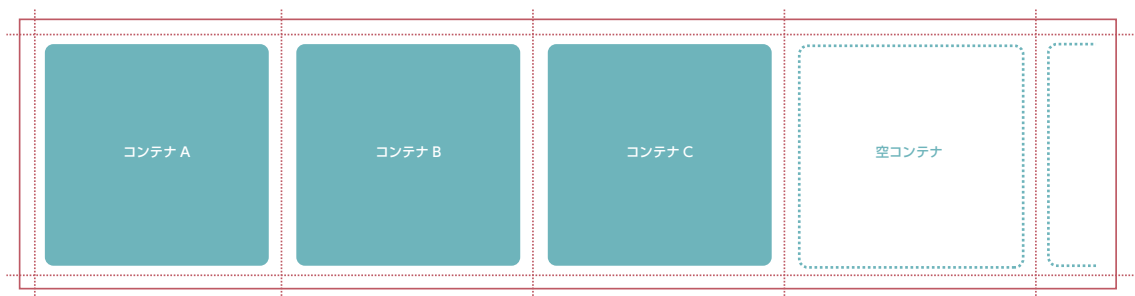
auto-fit を使用した場合

グリッドコンテナの幅より大きくなった場合、各要素が広がって表示される。



auto-fill を使用した場合

グリッドコンテナの幅より大きくなった場合、各要素が広がらず、空のコンテナが作成される。



minmax() 関数

グリッド内の領域寸法の範囲（最小値・最大値）を指定

これらを組み合わせて、解像度に合わせたレイアウトを表現します。

MDN - repeat()
[https://developer.mozilla.org/ja/docs/Web/CSS/repeat\(\)](https://developer.mozilla.org/ja/docs/Web/CSS/repeat())

MDN - minmax()
[https://developer.mozilla.org/ja/docs/Web/CSS/minmax\(\)](https://developer.mozilla.org/ja/docs/Web/CSS/minmax())

auto-fit , auto-fillを使ったGridレイアウトの記述

では、実際に先程のレイアウト例を元にコードで記述してみます。サンプルファイルを元にauto-fit , auto-fillを使ったグリッドレイアウトを作成し、解像度での変化を見てみましょう

sample7_grid_2.html - HTML

```
<h2 class="ttl">grid - autoFit の動作 </h2>
<div class="wrapFit">
  <section>
    <h2>page title 1</h2>
    <p>Lorem ipsum ...
  ~
</div>
<h2 class="ttl">grid - autoFill の動作 </h2>
<div class="wrapFill">
  <section>
    <h2>page title 4</h2>
    <p>Lorem ipsum ...
  ~
</div>
```

sample7_grid - CSS

```
.wrapFit{
  margin: 20px;
  padding: 20px;
  border: dotted 1px #999;
  display: grid;
  grid-template-columns: repeat(auto-fit,minmax(400px,1fr));
}

.wrapFill{
  margin: 20px;
  padding: 20px;
  border: dotted 1px #999;
  display: grid;
  grid-template-columns: repeat(auto-fill,minmax(200px,1fr));
}
```

repeat(繰り返し条件 , 最小大幅の領域)

auto-fit

400px~1fr

auto-fit

400px~1fr

repeat + auto-fit

最小400pxから残り全て広がるように設定。親要素が子カラム(400px x 個数)に入り切らない場合カラムを落として表示する。

auto-fill

200px~1fr

auto-fill

200px~1fr

auto-fill

200px~1fr

repeat + auto-fill

最小200pxで表示し、広がった分は空の要素を追加する。親要素が子カラム(200px x 個数)に入り切らない場合カラムを落として表示する。