

# float / flexbox / css grid

2022 年 6 月 3 日

## 概要

float / flexbox / css grid の違いについて学び、今時のレイアウト方法を学ぼう

## 目次

1	はじめに	2
1.1	読み間違えないでね . . . . .	2
1.2	注意 . . . . .	2
2	float はレイアウトを組むのには厄介！	2
2.1	VSCode のちょっとした機能 lorem ipsum . . . . .	2
2.2	id08-float.html . . . . .	2
2.3	child クラスにスタイルを当ててみよう . . . . .	3
2.4	float の欠点を治そう . . . . .	4
3	Flexbox は横並びに便利	5
3.1	id08-flexbox.html . . . . .	5
3.2	横並びにしてみよう . . . . .	6
3.3	要素の幅を統一してみよう . . . . .	6
3.4	縦の揃えをいろいろ試してみよう . . . . .	6
3.5	横の揃えをいろいろ試してみよう . . . . .	7
3.6	要素の並びを変えてみよう . . . . .	7
3.7	折り返しの位置を変えてみよう . . . . .	7
3.8	.flex-box を画面の高さ一杯に広げよう . . . . .	7
3.9	複数行の揃えを変更してみよう . . . . .	8
3.10	他にも機能があるけど... . . . . .	8
4	CSS Grid Layout は簡単！	8
4.1	id08-cssgrid.html . . . . .	8
4.2	トラックの指定をしよう . . . . .	9
4.3	アイテムの配置をしよう 方法 A：ライン番号でエリアを指定 . . . . .	9
4.4	アイテムの配置をしよう 方法 B：エリアに名前をつけて指定する方法 . . . . .	10
5	Grid Layout を試してみよう	10

# 1 はじめに

## 1.1 読み間違えないでね

ソースコード 1 読み間違えないでね

---

```
1 数字: 0123456789
2 小文字:abcdefghijklmnopqrstuvwxyz
3 大文字:ABCDEFGHIJKLMNOPQRSTUVWXYZ
4
5 1:イチ
6 l:小文字のエル
7 i:小文字のアイ
8 |:ビックリマーク
9 |:バーティカルバー。Shift と ¥ を押したものの。
10
11 0:ゼロ
12 o:小文字のオー
13 O:大文字のオー
14
15 .:ピリオド
16 ,:コンマ
```

---

## 1.2 注意

- これから出てくるソースコードには、左に「行番号」と呼ばれる番号が出てくるけど、入力する必要はないからね。

# 2 float はレイアウトを組むのには厄介！

## 2.1 VSCode のちょっとした機能 lorem ipsum

float に入る前に lorem ipsum について説明します。

出版、ウェブデザイン、グラフィックデザインなどの諸分野において使用されている典型的なダミーテキストのことを lorem ipsum(ローレム Ipsum) と言います。ダミーテキストのため、意味は全くありません。(一応、もとは古代ローマの政治家・哲学者キケロの『De finibus bonorum et malorum』(善と悪の究極について) という著作から取られている。)

vscode では、ipsum10, ipsum20, ipsum30 等と入力してリターンを押すことによって、その単語数分の文字を自動で入力してくれます。

以下の例では、ipsum10, ipsum20, ipsum30, ipsum50 を利用しています。

## 2.2 id08-float.html

どうなるか想像しながら入力してみよう。

---

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="UTF-8">
5         <title>float</title>
6     </head>
7     <body>
8         <div class="columns">
9             <div class="child">
10                Lorem ipsum dolor sit amet consectetur adipisicing elit.
11                Asperiores, eum?
12            </div>
13            <div class="child">
14                Lorem ipsum dolor sit amet consectetur, adipisicing elit.
15                Dolores facere numquam assumenda delectus totam nam
16                nisi accusamus est illum expedita!
17            </div>
18            <div class="child">
19                Lorem ipsum dolor sit amet consectetur adipisicing elit.
20                Debitis vel ducimus impedit omnis esse velit est, quo
21                at modi maxime tempore dicta labore beatae ipsa minima
22                maiores aut quisquam animi!
23            </div>
24            <div>
25                <p>
26                    Lorem ipsum dolor sit amet consectetur adipisicing
27                    elit. Dolor natus reprehenderit voluptate ab non
28                    molestias incidunt corrupti, distinctio, nobis
29                    quidem tempore rerum facere perferendis atque
30                    harum quisquam magni facilis fugit veniam possimus
31                    perspiciatis ea accusantium odit quia. Nam
32                    aliquid ea quod dolorem impedit asperiores! Amet
33                    quam sequi optio vitae consectetur!
34                </p>
35            </div>
36        </div>
37    </body>
38 </html>
```

---

## 2.3 child クラスにスタイルを当ててみよう

head タグの中に次の CSS を入れて見てみよう。入力する前にどんな感じになるか考えてみよう。

---

```
1 <style>
2     .child {
```

```
3         border: 1px solid black;
4         float: left;
5         width: 25%;
6         padding: 10px;
7         margin-right: 10px;
8     }
9     </style>
```

---

最後の文章は回り込んだね。

## 2.4 float の欠点を治そう

次の CSS を追加しよう

### ソースコード 4 CSS 部分 2

---

```
1     .columns::after {
2         content: "";
3         clear: both;
4         display: block;
5     }
```

---

このおまじないをすることで、一応レイアウトすることができるね。

最終系はこんな感じ。

### ソースコード 5 CSS 部分 1

---

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>float</title>
6     <style>
7         .child {
8             border: 1px solid black;
9             float: left;
10            width: 25%;
11            padding: 10px;
12            margin-right: 10px;
13        }
14        .columns::after {
15            content: "";
16            clear: both;
17            display: block;
18        }
19    </style>
20 </head>
21 <body>
22     <div class="columns">
23         <div class="child">
24             Lorem ipsum dolor sit amet consectetur adipisicing elit. Asperiores, eum?
```

```

25     </div>
26     <div class="child">
27         Lorem ipsum dolor sit amet consectetur, adipisicing elit. Dolores facere
            numquam assumenda delectus totam nam nisi accusamus est illum expedita
            !
28     </div>
29     <div class="child">
30         Lorem ipsum dolor sit amet consectetur adipisicing elit. Debitis vel
            ducimus impedit omnis esse velit est, quo at modi maxime tempore dicta
            labore beatae ipsa minima maiores aut quisquam animi!
31     </div>
32 </div>
33 <div>
34     <p>
35         Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor natus
            reprehenderit voluptate ab non molestias incidunt corrupti, distinctio
            , nobis quidem tempore rerum facere perferendis atque harum quisquam
            magni facilis fugit veniam possimus perspiciatis ea accusantium odit
            quia. Nam aliquid ea quod dolorem impedit asperiores! Amet quam sequi
            optio vitae consectetur!
36     </p>
37 </div>
38 </body>
39 </html>

```

---

## 3 Flexbox は横並びに便利

### 3.1 id08-flexbox.html

まずは、レイアウトは考えずに HTML, CSS を入力してみよう。lorem10, lorem20, lorem10, lorem5 を使ったよ。

---

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Flexbox</title>
6     <style>
7         .flex-box {
8             background-color: #eee;
9             padding: 10px;
10        }
11        .flex-item {
12            padding: 10px;
13            color: #fff;
14            margin: 10px;
15            border-radius: 5px;
16        }

```

```

17     .item1 {background-color: blue;}
18     .item2 {background-color: green;}
19     .item3 {background-color: darkblue;}
20     .item4 {background-color: darkred;}
21     </style>
22 </head>
23 <body>
24     <div class="flex-box">
25         <div class="flex-item item1">
26             1. Lorem ipsum dolor sit amet consectetur adipisicing elit. Cum, quaerat?
27         </div>
28         <div class="flex-item item2">
29             2. Lorem ipsum dolor sit amet consectetur adipisicing elit. Quia esse ad
                totam nostrum illum iusto deserunt molestiae iure at omnis.
30         </div>
31         <div class="flex-item item3">
32             3. Lorem ipsum dolor sit amet consectetur adipisicing elit. Non, fuga.
33         </div>
34         <div class="flex-item item4">
35             4. Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempora nisi
                qui sequi id debitis nemo eius fuga magni voluptate esse.
36         </div>
37     </div>
38 </body>
39 </html>

```

---

## 3.2 横並びにしてみよう

.flex-box に以下を追加

---

```
1     display: flex;
```

---

float と異なり、いい感じに並んだね。

item4 の div を削除して見ても勝手にいい感じにレイアウトされるね。実験した後元に戻しておこう。

## 3.3 要素の幅を統一してみよう

.flex-item に以下を追加

---

```
1     width: 15%;
```

---

右に空きがあるけど、綺麗に並ぶね。高さも揃ってるね。

## 3.4 縦の揃えをいろいろ試してみよう

.flex-box に以下を追加

---

```
1     align-items: flex-start;
```

---

flex-start を flex-end, center, stretch と変えて変化を見てみよう。最後 stretch にしておこう。

### 3.5 横の揃えをいろいろ試してみよう

.flex-box に以下を追加

```
1 justify-content: center;
```

center を flex-start, flex-end, space-between, space-around と変えて変化を見てみよう。最後、flex-start にしておこう。

### 3.6 要素の並びを変えてみよう

.flex-box に以下を追加

```
1 flex-direction: row-reverse;
```

row-reverse を row, column, column-reverse と変えて変化を見てみよう。

横並びって言うたのに、縦並びもできるね。考え方としては、1 次元 (1 方向) と考えていいよ。最後、row にしておこう。

### 3.7 折り返しの位置を変えてみよう

.flex-item の width を 30% にしておこう。折り返すね。

.flex-box に以下を追加

```
1 flex-wrap: wrap;
```

wrap を nowrap, wrap-reverse に変更してみよう。最後、wrap にしておこう。

### 3.8 .flex-box を画面の高さ一杯に広げよう

.flex-box に以下を追加

```
1 height: 100vh;
```

100vh はビューポート (見えているエリア) の高さを表すんだけど、なんかスクロールバー出ちゃうね。body にマージンが付いてるから、0 にしてみよう。

```
1 body {  
2     margin: 0;  
3 }
```

それでもスクロールバーが消えない... 理由は何かな？

そう、.flex-box の padding(外側の余白) が付いてるから。padding を 0 にするとスクロールバーは消えるね。これってめんどくさいよね。なので、width, height の計算方法を変えてみよう。

```
1 * {  
2     box-sizing: border-box;  
3 }
```

content-box   パディングとボーダーを幅と高さに含めない (初期値)  
border-box   パディングとボーダーを幅と高さに含める

おまじないのように、この CSS を入れておくと、いろいろ楽になるよ。

パディングとかなくなったので、3 段になっちゃったので、.flex-item の width を 40% にしておこう

### 3.9 複数行の揃えを変更してみよう

.flex-box に以下を追加

---

```
1 align-content: flex-start;
```

---

flex-start を flex-end, center, space-between, space-around, stretch と変えてみよう。

### 3.10 他にも機能があるけど...

横並び (一方向に並べる) ことが簡単にできることがわかったかな？逆に並べる意味はよくわからないと思う。

例えば、ニュースサイトがあったとして

- 古い記事から並べる
- 新しい記事から並べる

なんてことが、CSS だけで簡単にできるよ。

詳しく見てみたい人は、ここを見てみよう。 <https://www.sejuku.net/blog/56401>

## 4 CSS Grid Layout は簡単！

### 4.1 id08-cssgrid.html

---

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>CSS Grid Layout</title>
6     <style>
7         .container {
8             display: grid;
9         }
10        .item{color: white;}
11        .itemA{background-color: red;}
12        .itemB{background-color: green;}
13        .itemC{background-color: blue;}
14    </style>
15 </head>
16 <body>
17     <div class="container">
18         <div class="item itemA">A</div>
```



```
19         <div class="item itemB">B</div>
20         <div class="item itemC">C</div>
21     </div>
22 </body>
23 </html>
```

---

## 4.2 トラックの指定をしよう

.container の中に追加

```
1         grid-template-rows: 100px 50px;
2         grid-template-columns: 150px 1fr;
```

---

1 行目の高さが 100px, 2 行目の高さが 50px;

1 列目の幅が 150px, 2 列目の幅は残り幅 (fr: fraction: 分数) という意味です。  
セルに順番に割り当てられましたね。

## 4.3 アイテムの配置をしよう 方法 A：ライン番号でエリアを指定

id08-cssgrid.html をコピーして id08-cssgrid-a.html として作業しましょう。

grid-row アイテムが占める行のライン番号を「/」で区切り指定する

grid-column アイテムが占める列のライン番号を「/」で区切り指定する

A を左に縦に使い、B,C を右に上下に並べて見ましょう。.itemA に追加

```
1         grid-row: 1/3;
```

---

これで、目的は達せられましたが、ちゃんと全てを正しく指定した場合には以下のようになります。

```
1     .itemA{
2         grid-row: 1/3;
3         grid-column: 1/2;
4         background-color: red;
5     }
6     .itemB{
7         grid-row: 1/2;
8         grid-column: 2/3;
9         background-color: green;}
10    .itemC{
11        grid-row: 2/3;
12        grid-column: 2/3;
13        background-color: blue;
14    }
```

---

もし、エリアが被った場合にはどうなるのでしょうか？

.itemC の grid-row を 1/3 にしたり,grid-column を 1/3 にして見たりしましょう。

HTML の後ろに記載されてるものが上に乗っかる形になる様です。

## 4.4 アイテムの配置をしよう 方法 B：エリアに名前をつけて指定する方法

id08-cssgrid.html をコピーして id08-cssgrid-a.html として作業しましょう。

.container に以下を追加して見ましょう。

```
1      grid-template-areas:  
2          "areaA areaB"  
3          "areaA areaC";
```

これで各セルに名前をつけることができました。

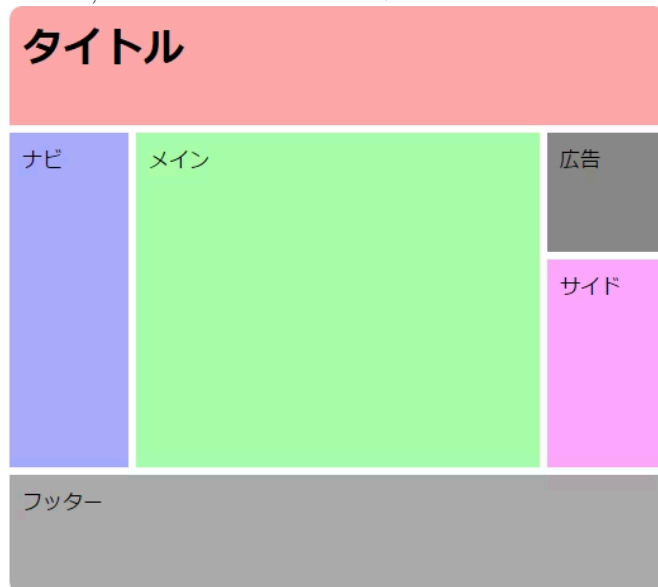
それでは、次にならって、.itemB, .itemC にも grid-area を設定してください。

```
1      .itemA{  
2          grid-area: areaA;  
3          background-color: red;  
4      }
```

簡単でしょ？

## 5 Grid Layout を試してみよう

方法 A, 方法 B で試してみよう。



時間余った人は、メインの中に Flexbox を使って、記事を 3 つ横に並べてみよう。

参考 <https://qiita.com/kura07/items/e633b35e33e43240d363>

以上