angular flex layoutについて

小西 孝海

# 基本

angularのflex-layoutの基本的な考え方について図 1.1を元に説明する。

flex-container(fxLayout)

flex-item

(fxFlex)

flex-item

(fxFlex)

flex-item

(fxFlex)

FlexGap

FlexGap

FlexOrder

図 1.1 レイアウトの定義

まず、いくつかのアイテム、例えばcardやformといった要素のことを図 1.1ではflex-itemとして記している。

そしていくつかのflex-itemを一つにまとめたものをflex-containerとしている。

angular flex layoutでは、主に4つの機能に分かれている。

**fxLayout**

fxLayoutは、flex-itemを縦並びに表示するか、横並びに表示するかを制御する。

**fxFlex**

fxFlexは、flex-itemの表示幅に関する制御を行う。

**fxLayoutGap**

fxLayoutGapでは、隣接するflex-item同士のマージンに関する制御を行う。

**fxFlexOrder**

fxFlexOrderでは、flex-container内の表示順序を制御する。

以降では、これらの詳細について説明する。

なお、まずは理解して動かす前提のため、細かな部分については記載していないものもある。詳細について知りたい場合は以下のURLを参考にすること。

**[angular wiki]**

<https://www.angularjswiki.com/flexlayout/>

# fxFlex

fxFlexはfxLayoutコンテナ内のflex-itemで使用する必要がある。

fxFlexは、レイアウトの主軸に沿って要素(flex-item)のサイズを変更する役割を持つ。

## flex-grow

flex-growは、表示画面に十分なスペースがある場合に、同じflex-container内の残りのflex-itemと比較してどれだけ大きくなるかを指定する。

grow指定をした場合と、growを指定していないときのカードの表示例を図 2.1に、表示の際のHTMLを図 2.2に記す。



図 2.1 growの表示例

|  |
| --- |
| <h2>grow指定あり</h2><div class="flex-container" fxLayout="row">    <mat-card class="child-1" fxFlex="grow">1. One</mat-card>    <mat-card class="child-2" fxFlex="grow">1. Two</mat-card>    <mat-card class="child-3" fxFlex="grow">1. Three</mat-card></div><h2>grow指定なし</h2><div class="flex-container" fxLayout="row">    <mat-card class="child-1">1. One</mat-card>     <mat-card class="child-2">1. Two</mat-card>     <mat-card class="child-3">1. Three</mat-card></div> |

図 2.2 サンプルコード

growの指定がある場合、flex-item(mat-card)が画面一杯に対して3等分されて表示される。

growの指定がない場合、flex-itemはデフォルトの大きさで表示される。

flex-growに指定できる値は、0または1、または任意の整数値を指定できる。指定できる値と、値の意味を表 2.1に記す。

表 2.1 flex-growに指定できる値と意味

|  |  |
| --- | --- |
| 指定できる値 | 意味 |
| 0 | 伸縮しない。 |
| 1 | **デフォルト値。**同じ行にある他のflex-itemと同じサイズになる。 |
| 任意の整数 | 同じ行に「flex-grow:1」がある他の要素のサイズのn倍になる。 |

## flex-shrink

flex-shrinkは、表示画面に十分なスペースがない場合に、同じflex-container内の残りのflex-itemと比較してどれだけ縮小するかを指定する。

表 2.2 flex-shrinkに指定できる値と意味

|  |  |
| --- | --- |
| 指定できる値 | 意味 |
| 0 | 伸縮しない。 |
| 1 | **デフォルト値。**同じ行にある他のflex-itemと同じサイズになる。 |
| 任意の整数 | 同じ行に「flex-grow:1」がある他の要素のサイズの1/nになる。 |

## flex-basis

flex-basisは、flex-itemの初期のデフォルトのサイズを指定する。

flex-layoutでは、flex-basisで初期サイズだけを指定すると、flex-itemは拡大または縮小しない。

flex-basisには、次の単位のいずれかを指定できる。

1. ピクセル
2. パーセンテージ
3. calc()
4. em
5. vw
6. vh
7. aliases

単位が指定されていない場合、デフォルトの単位はパーセンテージである。

fxFlex=”30”と指定した場合は30%の大きさで表示される。

上記7のエイリアスに指定できる値とCSS上の意味を表 2.3に記す。

表 2.3　エイリアスに指定できる値とCSSの関係

|  |  |
| --- | --- |
| alias | CSS |
| grow | flex: 1 1 100% |
| initial | flex: 0 1 auto |
| auto | flex: grow shrink 100% |
| none | flex 0 0 auto |
| nogrow | flex: 0 1 auto |
| noshrink | flex:1 0 auto |

calcを指定する場合は、以下のように指定する。

|  |
| --- |
| <div fxFlex=”2 1 calc(10px + 100px)”></div> |

## fxFlexの記法

fxFlexでは、flex-itemにflex-grow、flex-shirnk、flex-basisを記載する。

|  |
| --- |
| <div fxLayout=”row”> <div fxFlex=”<flex-grow> <flex-shrink> <flex-basic>”></div></div> |

## fxFlex短縮形

fxFlexを記載するとき、短縮した指定方法と、すべて記載する方法がある。

短縮形ではflex-basisだけ指定できる。

|  |
| --- |
| <div fxLayout=”row”> <div fxFlex=”<flex-basis>”></div></div> |

# fxLayout

flex-itemを縦並びに表示するか、横並びに表示するかを指定する機能である。

そのため、fxLayoutは、flex-containerで使用する必要がある。

横並びでflex-itemを表示する場合、fxLayoutの属性にrowを、縦並びに表示する場合、columnを指定する。

fxLayoutにrow、またはcolumnを指定していない場合、rowが仮定される。

fxLayoutのサンプルコードを図 3.1に、表示結果は図 3.2に記す。

|  |
| --- |
| <h2>横並び</h2><div class="flex-container" fxLayout="row">    <mat-card class="child-1" fxFlex="25%">1. One</mat-card>     <mat-card class="child-2" fxFlex="25%">1. Two</mat-card>     <mat-card class="child-3" fxFlex="25%">1. Three</mat-card>    </div><h2>縦並び</h2><div class="flex-container" fxLayout="column" style="width: 75%">    <mat-card class="child-1">1. One</mat-card>     <mat-card class="child-2">1. Two</mat-card>     <mat-card class="child-3">1. Three</mat-card>    </div> |

図 3.1 fxLayoutのサンプルコード



図 3.2 表示結果

なお、縦並びの場合、各要素は画面いっぱいに広がる。そのため、図 3.1のサンプルコードでは「style="width: 75%"」を指定してコンテナのサイズ自体を画面の75%にし、横並びの表示と同じ幅になるように設定している。

# fxLayoutGap

fxLayoutGapは、flex-container内のflex-item間の表示間隔を指定するために使用する。fxLayoutGapはオプションであり、省略できる。

fxLayoutGapの指示は、親コンテナ、つまりflex-containerに追加する必要がある。

fxLayoutGapのサンプルコードを図 4.1に、fxLayoutGapの表示例を図 4.2に記す。

|  |
| --- |
| <h2>FxLayoutGapなし</h2><div class="flex-container" fxLayout="row">    <mat-card class="child-1" fxFlex="25%">1. One</mat-card>     <mat-card class="child-2" fxFlex="25%">1. Two</mat-card>     <mat-card class="child-3" fxFlex="25%">1. Three</mat-card>    </div><h2>FxLayoutGapあり</h2><div class="flex-container" fxLayout="row" fxLayoutGap="20px">    <mat-card class="child-1" fxFlex="25%">1. One</mat-card>     <mat-card class="child-2" fxFlex="25%">1. Two</mat-card>     <mat-card class="child-3" fxFlex="25%">1. Three</mat-card>    </div> |

図 4.1 fxLayoutGapのサンプルコード



図 4.2 fxLayoutGapの表示例

# fxFlexOrder

fxFlexOrderは、flex-container内のフレックス子アイテムに追加する必要がある。

fxFlexOrderは、要素の位置の順序を定義する。

fxFlexOrderのサンプルコードを図 5.1に、表示結果を図 5.2に記す。

|  |
| --- |
| <h2>FxFlexOrderなし</h2><div class="flex-container" fxLayout="row">    <mat-card class="child-1" fxFlex="25%">1. One</mat-card>    <mat-card class="child-2" fxFlex="25%">2. Two</mat-card>    <mat-card class="child-3" fxFlex="25%">3. Three</mat-card></div><h2>FxFlexOrderあり</h2><div class="flex-container" fxLayout="row">    <mat-card class="child-1" fxFlex="25%" fxFlexOrder="3">1. One</mat-card>    <mat-card class="child-2" fxFlex="25%" fxFlexOrder="2">2. Two</mat-card>    <mat-card class="child-3" fxFlex="25%" fxFlexOrder="1">3. Three</mat-card></div> |

図 5.1 fxFlexOrderのサンプルコード



図 5.2 fxFlexOrderの表示例

－以上－