Web プログラミング

第八回：CSS(5)

CSS(5) - 要素の重ね順

```
.greenbox{
    width: 600px;
    height: 200px;
    background-color: green;
    position: absolute;
    left: 30px;
    right: 30px;
}
.redbox{
    width: 200px;
    height: 400px;
    background-color: red;
}
```

position などを利用して位置を変えると、
他の要素と重なることがあります。

CSS(5) - 要素の重ね順

```
.greenbox{
    width: 600px;
    height: 200px;
    background-color: green;
    position: absolute;
    left: 30px;
    right: 30px;
    z-index: 1;
}
.redbox{
    width: 200px;
    height: 400px;
    background-color: red;
}
```

z-index プロパティを指定することで、
その要素の重ね順を決めることができます。
重ね順は、数値が大きいほど手前に置かれます。
（初期値：0）

CSS(5) - ファーストビューのブラッシュアップ

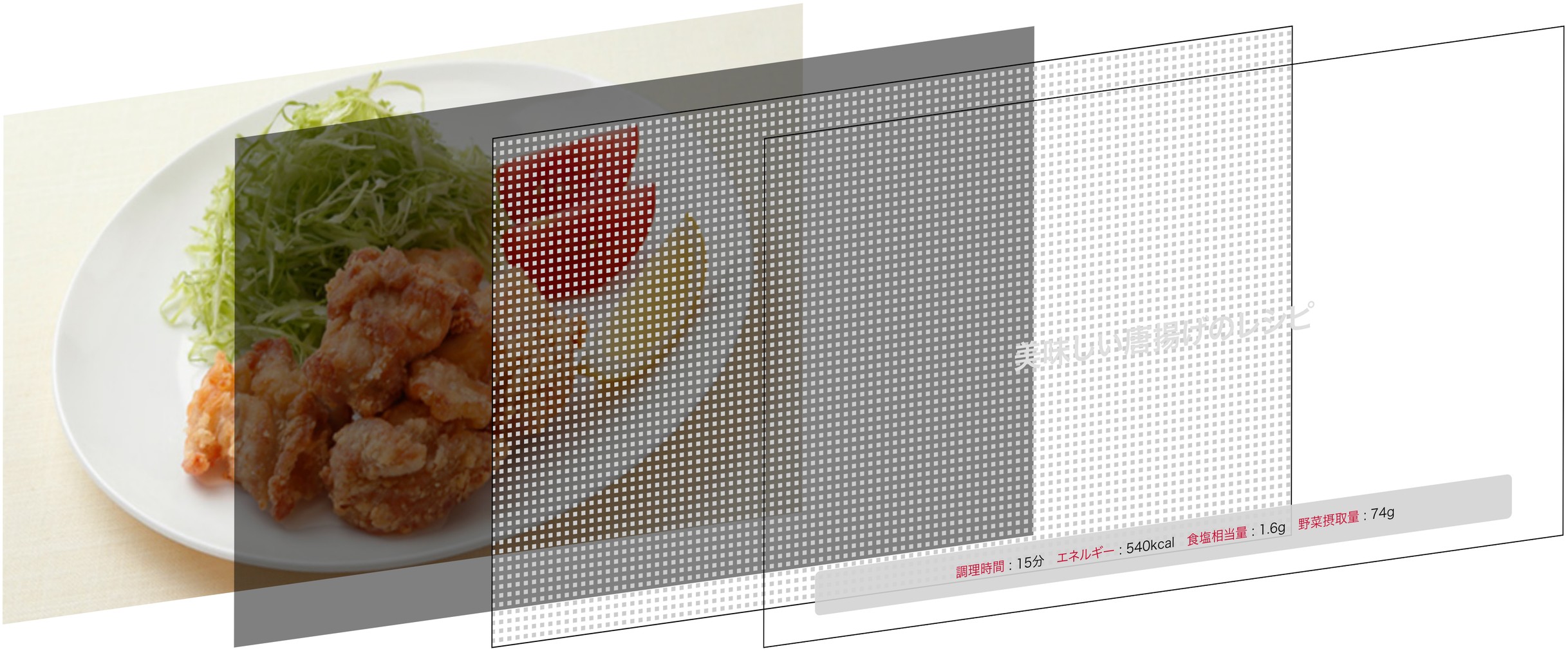
ファーストビューのデザインを、
position:absolute を使ってお手本のように改良します。

CSS(5) - ファーストビューのブラッシュアップ

現在、背景画像を敷いたブロックの中に
コンテンツが入っている形です

CSS(5) - ファーストビューのブラッシュアップ

コンテンツと背景画像の間に、
半透明の黒で塗りつぶしたブロックと、ドットの背景を敷き詰めたブロックを差し込みます。



CSS(5) - ファーストビューのブラッシュアップ

半透明の黒背景用のブロックを作成し、スタイルを当てます。

HTML

```
<div class="firstview">
    <h2 class="title"> 美味しい唐揚げのレシピ </h2>
    <div class="information">
        ... 省略 ...
    </div>
    <div class="black_layer"></div>
</div>
```

CSS

```
.black_layer{
    width: 100%;
    height: 100%;
    background-color: rgb(0,0,0,0.3);
    position: absolute;
    left: 0px;
    top: 0px;
}
```

CSS(5) - ファーストビューのブラッシュアップ

ドット背景用のブロックを作成し、スタイルを当てます。

HTML

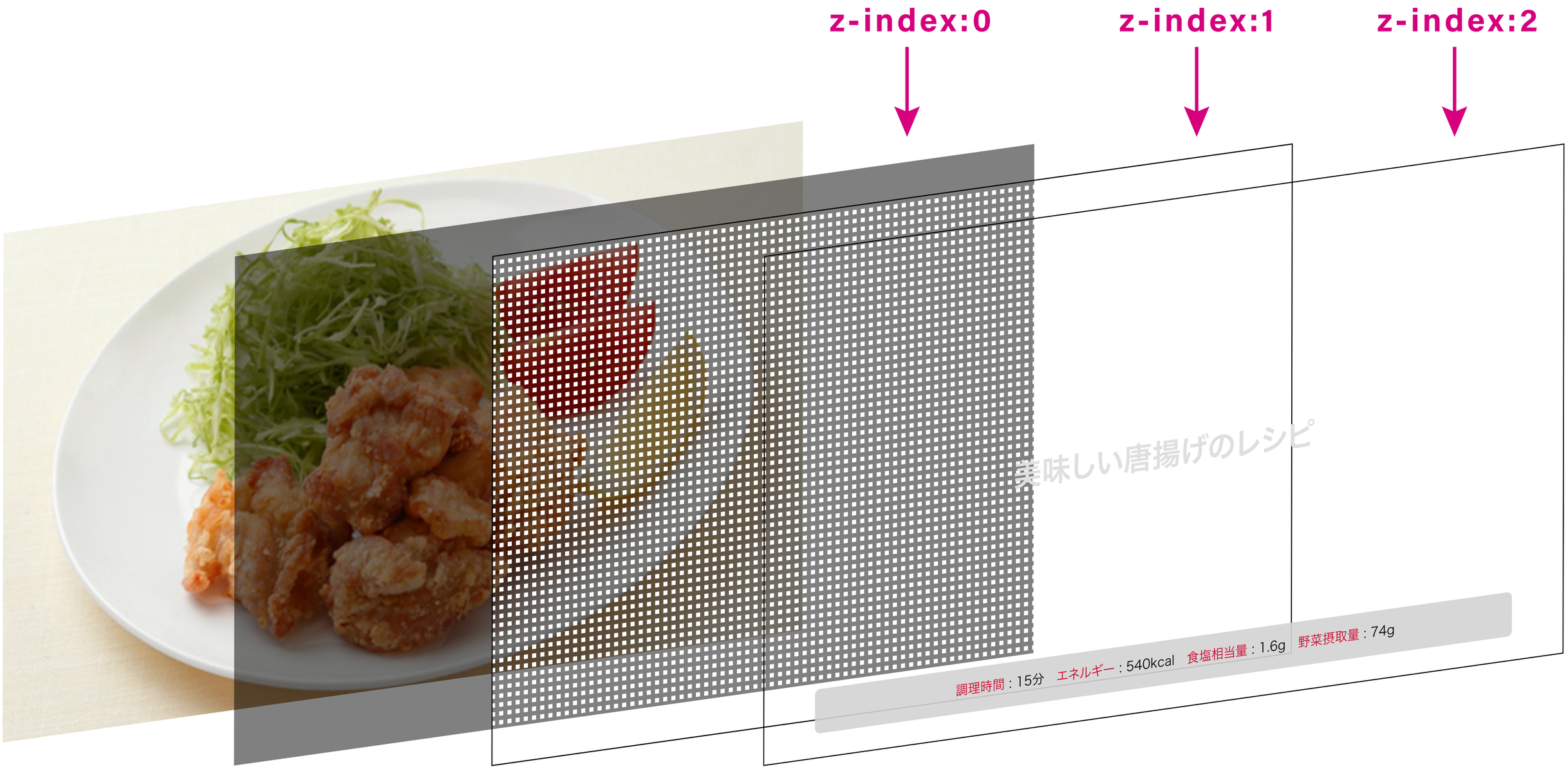
```
<div class="firstview">
    <h2 class="title"> 美味しい唐揚げのレシピ </h2>
    <div class="information">
        ... 省略 ...
    </div>
    <div class="black_layer"></div>
    <div class="dot_layer"></div>
</div>
```

CSS

```
.dot_layer{
    width: 100%;
    height: 100%;
    background-image: url(../images/hero_bg_dot.png);
    position: absolute;
    left: 0px;
    top: 0px;
}
```

CSS(5) - ファーストビューのブラッシュアップ

最後に z-index でブロックの重なり順を調整します



CSS(5) - フレックスボックスについて

通常のフローレイアウトでは、ボックスは縦に並びます

HTML

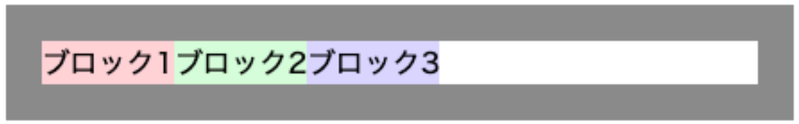
```
<div class="flexbox">
   <div class="block block_1">ブロック 1</div>
   <div class="block block_2">ブロック 2</div>
   <div class="block block_3">ブロック 3</div>
</div>
```

CSS

```
.flexbox{
   width: 400px;
   border: 20px solid rgb(0,0,0,0.5);
}
.block_1{ background-color: rgb(255,0,0,0.2); }
.block_2{ background-color: rgb(0,255,0,0.2); }
.block_3{ background-color: rgb(0,0,255,0.2); }
```

CSS(5) - フレックスボックスについて

display:flex を指定してフレックスボックスにすると、
その内部でのブロックの並ぶ方向を変化させることができます。



HTML

```
<div class="flexbox">
    <div class="block block_1">ブロック 1</div>
    <div class="block block_2">ブロック 2</div>
    <div class="block block_3">ブロック 3</div>
</div>
```

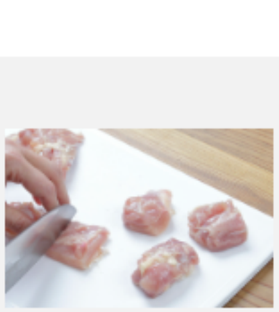
CSS

```
.flexbox{
    width: 400px;
    border: 20px solid rgb(0,0,0,0.5);
    display: flex;
}
.block_1{ background-color: rgb(255,0,0,0.2); }
.block_2{ background-color: rgb(0,255,0,0.2); }
.block_3{ background-color: rgb(0,0,255,0.2); }
```

CSS(5) - フレックスボックスについて

鶏もも肉	250g
しょうゆ	大さじ1 ※
酒	大さじ1 ※

1



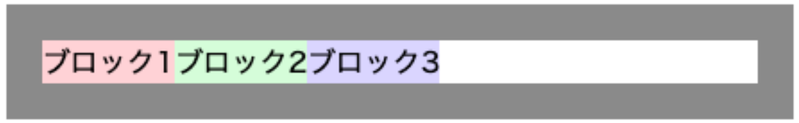
鶏肉を切る

鶏肉は脂身を取り除き、大きめの一口大（20〜30g）に切る。

display:flex を上手に使うと、
これらのような横並びを含むレイアウトを
実装することができます。

CSS(5) - フレックスボックス詳細

フレックスボックス要素とその直下のブロック要素に
様々な設定をすることで、レイアウトを複雑に変化させることができます。



半透明のプロパティは、display:flex のときのデフォルトで設定されるプロパティ。

```
.flexbox{
    display: flex;
    flex-direction: row;
    flex-wrap: nowrap;
    justify-content: flex-start;
    align-items: stretch;
    gap: 0px 0px;
}
.block{
    flex-basis: auto;
    flex-grow: 0;
    flex-shrink: 1;
}
```

CSS(5) - フレックスボックス（親要素）のプロパティ

flex-direction

ボックスの並ぶ方向（主軸）を指定する

```
.flexbox{
   flex-direction: row;
}
```

flex-direction: row

ボックスを横方向に並べる。デフォルト

```
.flexbox{
   flex-direction: column;
}
```

flex-direction: column

ボックスを縦方向に並べる。

```
.flexbox{
   flex-direction: row-reverse;
}
```

flex-direction: row-reverse

ボックスを横方向（逆順）に並べる。

```
.flexbox{
   flex-direction: column-reverse;
}
```

flex-direction: row

ボックスを縦方向（逆順）に並べる。

CSS(5) - フレックスボックス（親要素）のプロパティ

flex-wrap

内包する要素が溢れたとき、折り返すかどうかを指定する

```
.flexbox{
    flex-wrap: nowrap;
}
```

flex-wrap: nowrap

ボックスを折り返さない。デフォルト

```
.flexbox{
    flex-wrap: wrap;
}
```

flex-wrap: wrap

ボックスを折り返す。

```
.flexbox{
    flex-wrap: wrap-reverse;
}
```

flex-wrap: wrap-reverse

ボックスを逆方向に折り返す。

CSS(5) - フレックスボックス（親要素）のプロパティ

justify-content

フレックスボックスの並び方向に対して、内包する要素の揃え位置を指定する

```
.flexbox{ justify-content: flex-start; }
```

justify-content: flex-start
開始位置に揃える。デフォルト

```
.flexbox{ justify-content: flex-end; }
```

justify-content: flex-end
終了位置に揃える。

```
.flexbox{ justify-content: center; }
```

justify-content: center
中央に揃える。

```
.flexbox{ justify-content: space-between; }
```

justify-content: space-between
余ったスキマを子要素の間に均等に割り振る

```
.flexbox{ justify-content: space-around; }
```

justify-content: space-around
余ったスキマを子要素の左右に均等に割り振る

```
.flexbox{ justify-content: space-evenly }
```

justify-content: space-evenly
余ったスキマをボックスの内側と子要素の間に均等に割り振る

CSS(5) - フレックスボックス（親要素）のプロパティ

align-items
子要素どうしの揃え位置を指定する

```
.flexbox{
    align-items: stretch;
}
```

align-items: stretch

子要素のサイズを揃えて並べる。デフォルト

```
.flexbox{
    align-items: flex-start;
}
```

align-items: flex-start

子要素どうしを開始位置に揃えて並べる

```
.flexbox{
    align-items: flex-end;
}
```

align-items: flex-end

子要素どうしを終了位置に揃えて並べる

CSS(5) - フレックスボックス（親要素）のプロパティ

gap

子要素と子要素の間の間隔を指定する

```
.flexbox{
    gap: 10px 20px;
}
```

値を 2 つ指定する。
一つ目の値（10px）は、子要素と子要素の縦のスキマ
二つ目の値（20px）は、子要素と子要素の横のスキマ

CSS(5) - フレックスボックスの子要素のプロパティ

flex-basis

子要素の主軸方向のサイズ。

flex-direction が row（横）の場合、横幅

flex-direction が column（縦）の場合、高さになる

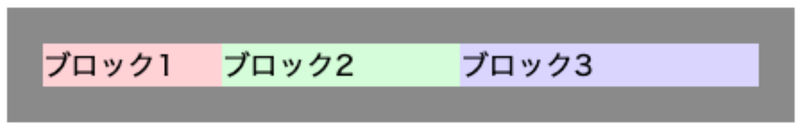
```
.box{
    flex-basis: 100px;
}
```

CSS(5) - フレックスボックスの子要素のプロパティ

flex-grow

主軸方向に対して余りがある場合、

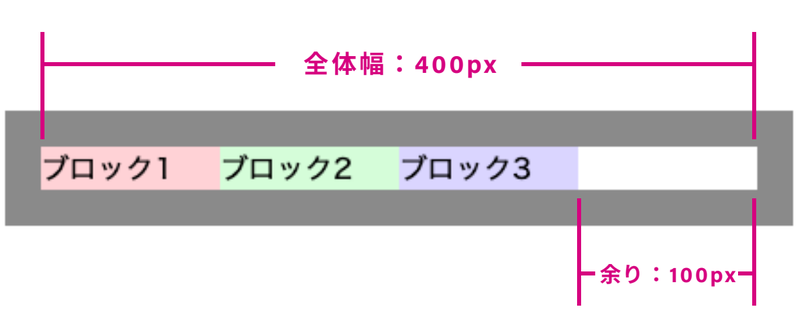
その余りに対してどの程度拡大するかの指定




```
.block{ flex-basis: 100px; }
.block_1{ flex-grow: 0; }
.block_2{ flex-grow: 1; }
.block_3{ flex-grow: 2; }
```

CSS(5) - flex-grow の計算方法

親フレックスボックの横幅が 400px あり、
各ブロックのサイズ（flex-basis）が 100px のとき、
余りは 100px あることになる。




```
.block{ flex-basis: 100px; }
```

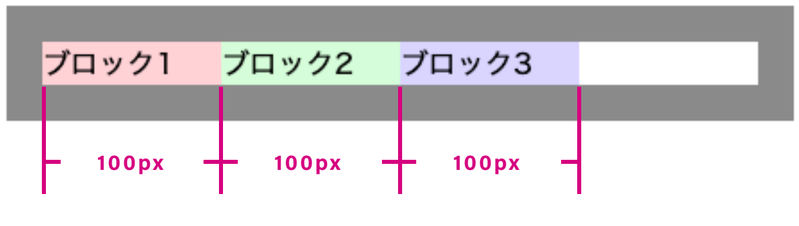
CSS(5) - flex-grow の計算方法

この余った 100px を、子要素の flex-glow の比率で分配して拡大する。

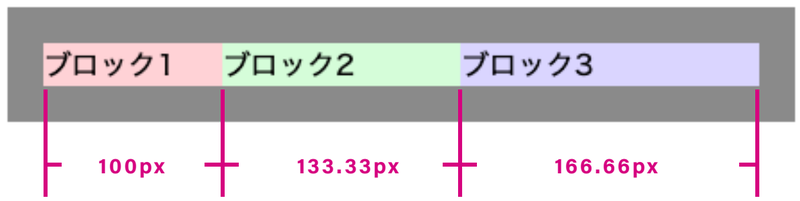
block_1 は 0、block_2 は 1、block_3 は 2 とすると、

0 : 33.33 : 66.66 となるので、その分だけ flex-basis に加算する。

flex-grow 適用前



flex-grow 適用後

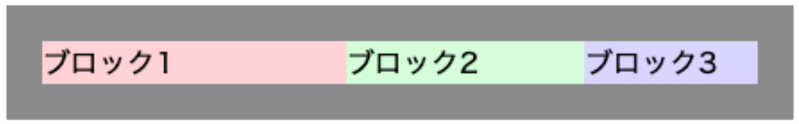



```
.block{ flex-basis: 100px; }
.block_1{ flex-grow: 0; }
.block_2{ flex-grow: 1; }
.block_3{ flex-grow: 2; }
```

CSS(5) - フレックスボックスの子要素のプロパティ

flex-shrink

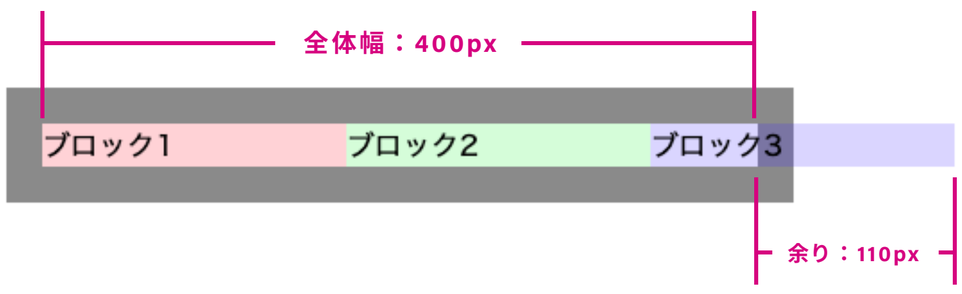
主軸方向のサイズに対して、子要素のサイズの方が大きい場合、
溢れたサイズに対してどの程度縮小するかの指定




```
.block{ flex-basis: 170px; }
.block_1{ flex-shrink: 0; }
.block_2{ flex-shrink: 1; }
.block_3{ flex-shrink: 2; }
```

CSS(5) - flex-shrink の計算方法

親フレックスボックの横幅が 400px あり、
各ブロックのサイズ（flex-basis）が 170px のとき、
110px 親ボックスからはみ出ることになる。



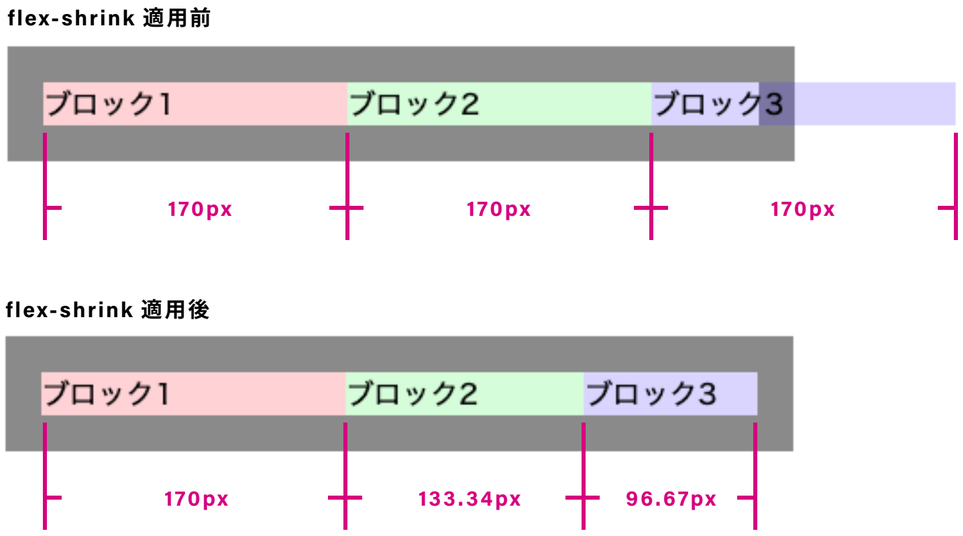

```
.block{ flex-basis: 170px; }
.block_1{ flex-shrink: 0; }
.block_2{ flex-shrink: 0; }
.block_3{ flex-shrink: 0; }
```

CSS(5) - flex-shrink の計算方法

この余った 110px を、子要素の flex-shrink の比率で分配して縮小する。

block_1 は 0、block_2 は 1、block_3 は 2 とすると、

0 : 36.66 : 73.33 となるので、その分だけ flex-basis から減算する。




```
.block{ flex-basis: 170px; }
.block_1{ flex-shrink: 0; }
.block_2{ flex-shrink: 1; }
.block_3{ flex-shrink: 2; }
```