

変態的 CSS トリック



Kite

KITERETZ inc. CEO & Funder
f kite.koga t ixkaito i ixkaito

-  Web & graphic designer
-  Programmer
-  WordPress core contributor
-  Ruby on Rails contributor
-  Wocker developer
-  Bathe developer
-  Frasco developer

前提

- Flexbox を使わない
- calc() を使わない

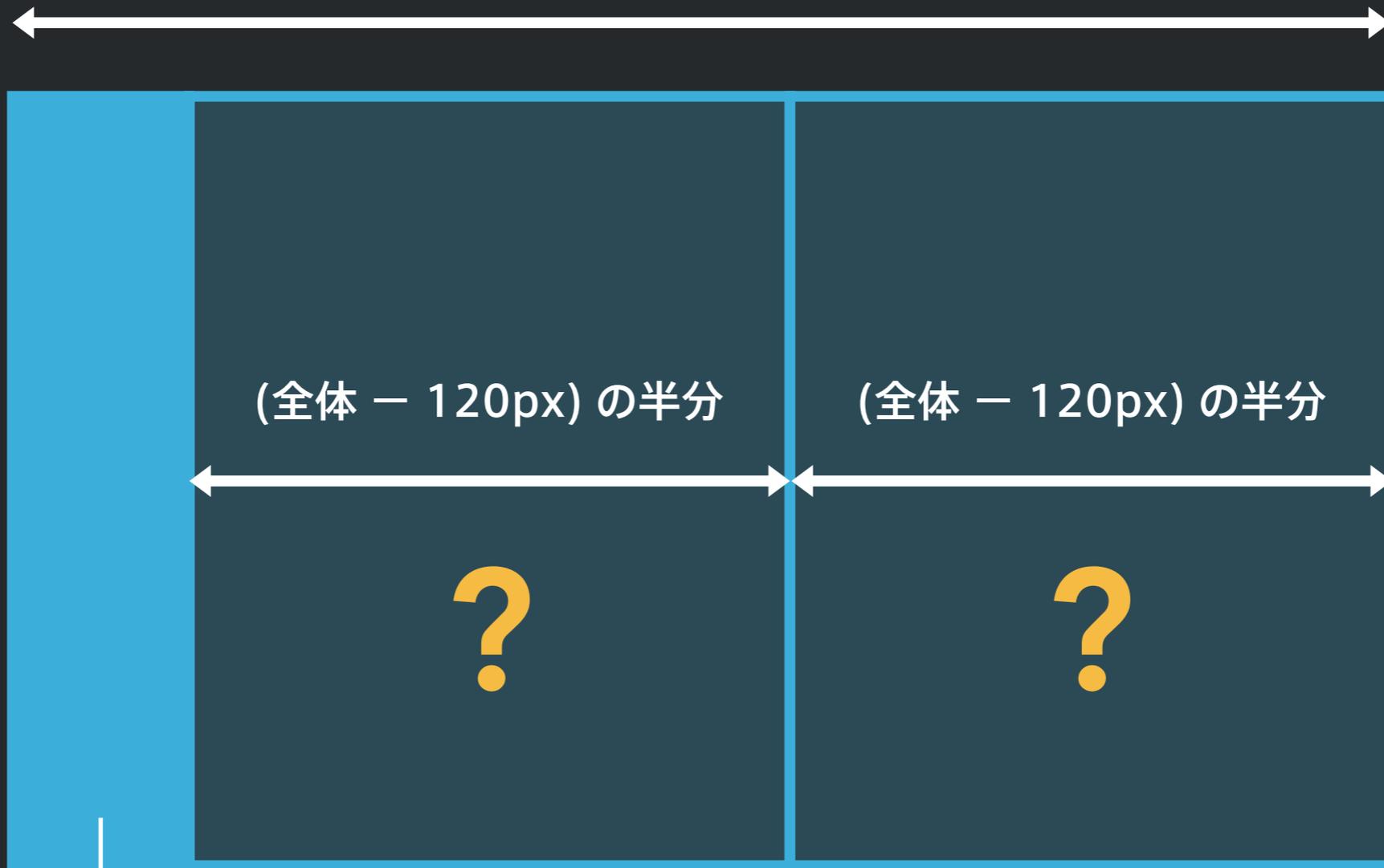
未対応ブラウザでレイアウトへの影響が大きいため

変態度：初級

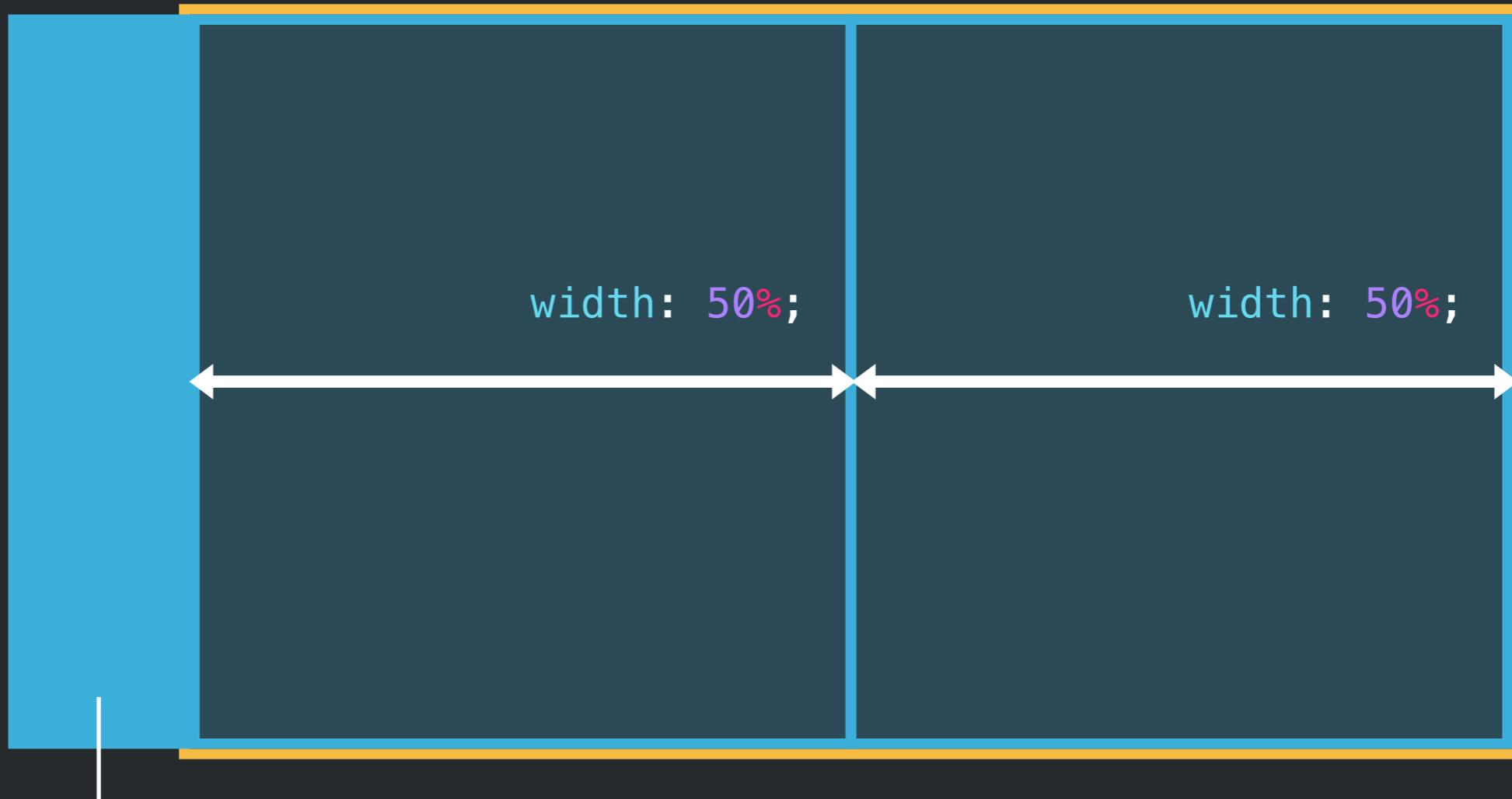
固定幅とパーセンテージの組み合わせ

	Sessions	Workshops
10:00	<p>開場・受付 [10:00]</p> <p>オープニングセレモニー / 細谷 崇 [10:30-10:40]</p> <p>基調講演：「子供達にプログラミングをサポートする上で大事なポイント」 / 阿部 和広 [10:40-11:20]</p>	<p>開場・受付 [10:00]</p> <p>Scratchって何？初めてのCoderDojo! -オトナも子どももニンジャになろう- [10:40-12:10]</p> <p>事前のエントリー（申込）とPCを持参頂く必要があります。 定員：大人6名、子供6名（大人だけ、子供だけでも参加可能です）</p> <p>Entry</p>
11:00	<p>学生の道場運営者に聞く！ / 宮島 衣瑛、東 和樹、安川 要平 [11:25-11:45]</p>	

レスポンス

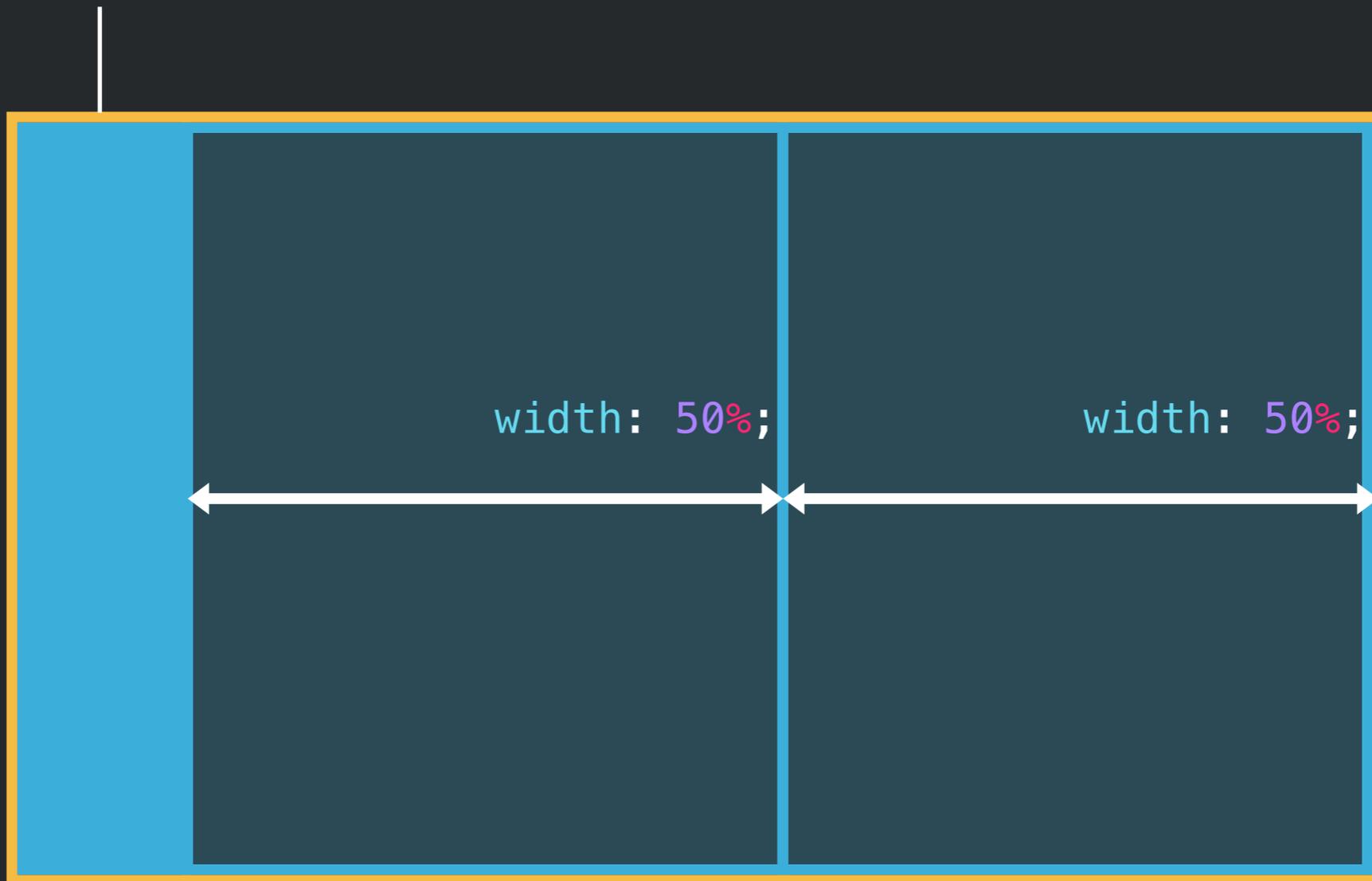


width: 120px;



`margin-left: -120px;` // 固定幅分のネガティブマージンを入れる

```
padding-left: 120px; // 固定幅分のパディングを入れる
```

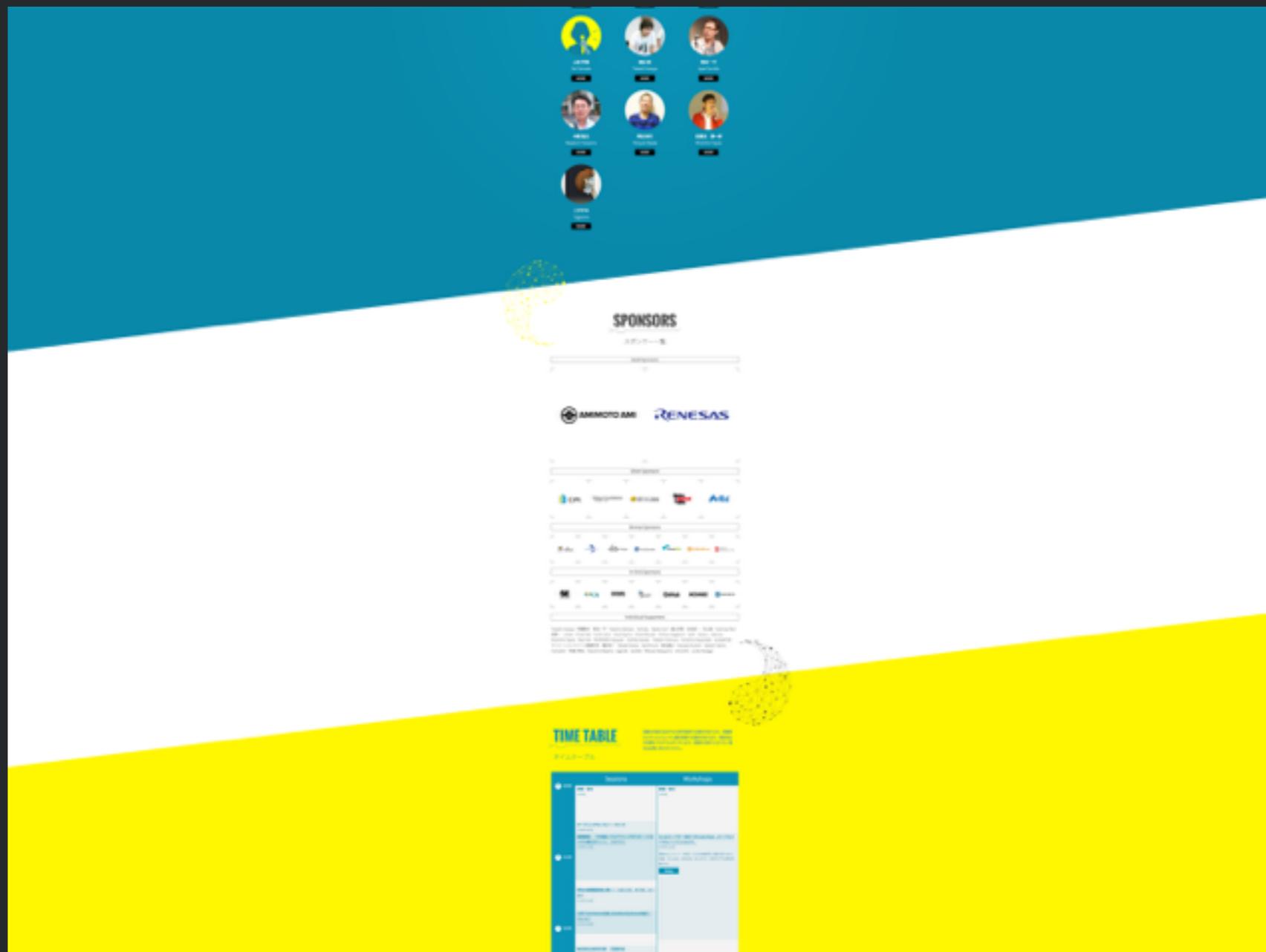


Demo: <https://ixkaito.github.io/percent-with-fixed-width/>

Source: <https://github.com/ixkaito/percent-with-fixed-width>

変態度：中級

異なる背景色の斜めデザイン



仕様



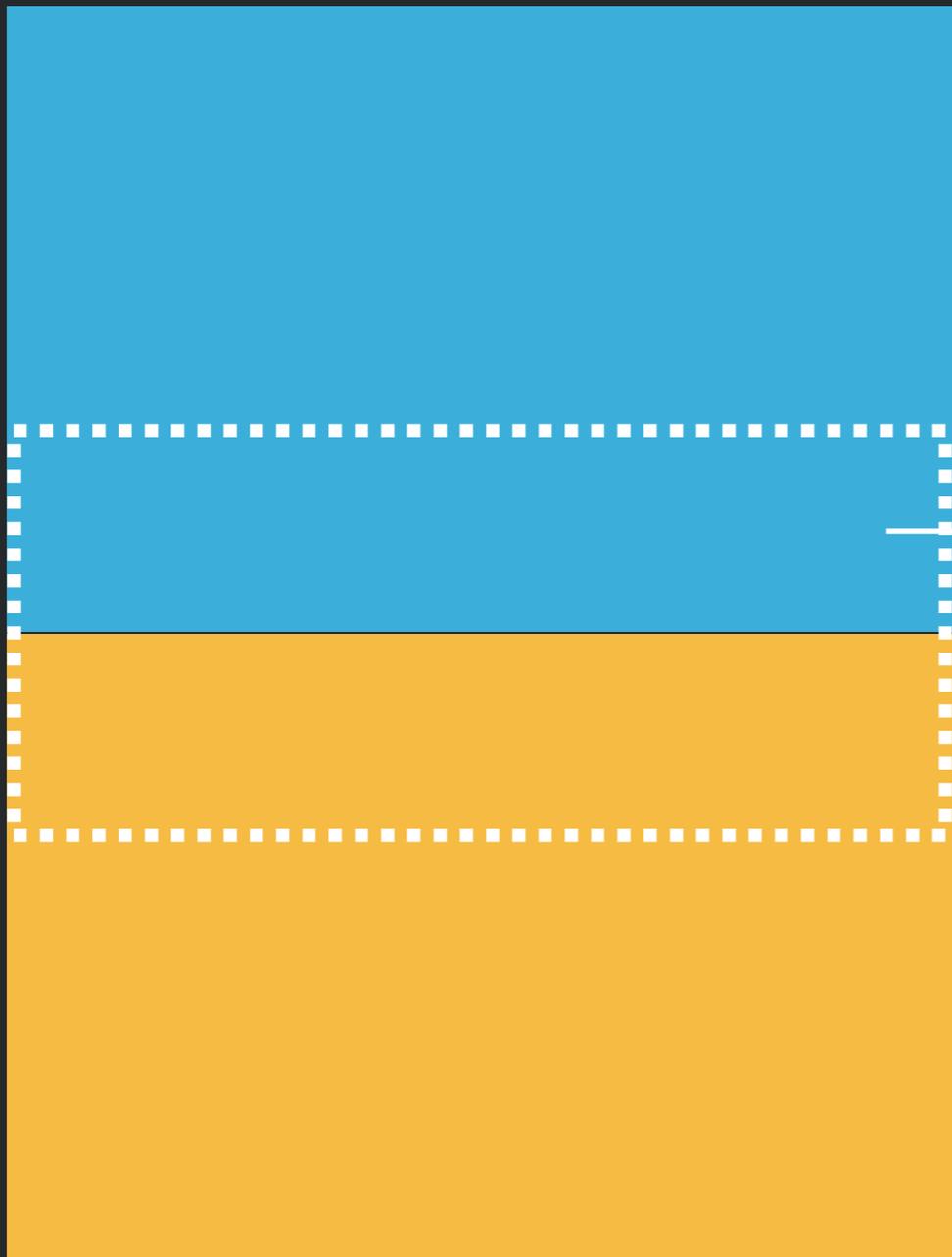
水平

水平

レスポンス



普通に `skew` をかけると...



```
// 背景色を半分ずつ上下の要素に合わせる  
background: linear-gradient(  
  blue 50%, yellow 50%  
);  
position: relative;  
z-index: 0; // 0 以上
```



```
// 擬似要素に skew をかける
:before {
  background: white;
  content: "";
  position: absolute;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: -1; // -1 以下
  transform: skewY(-7deg);
}
```

Demo: <https://ixkaito.github.io/skew-with-different-colors/>

Source: <https://github.com/ixkaito/skew-with-different-colors>

変態度：上級

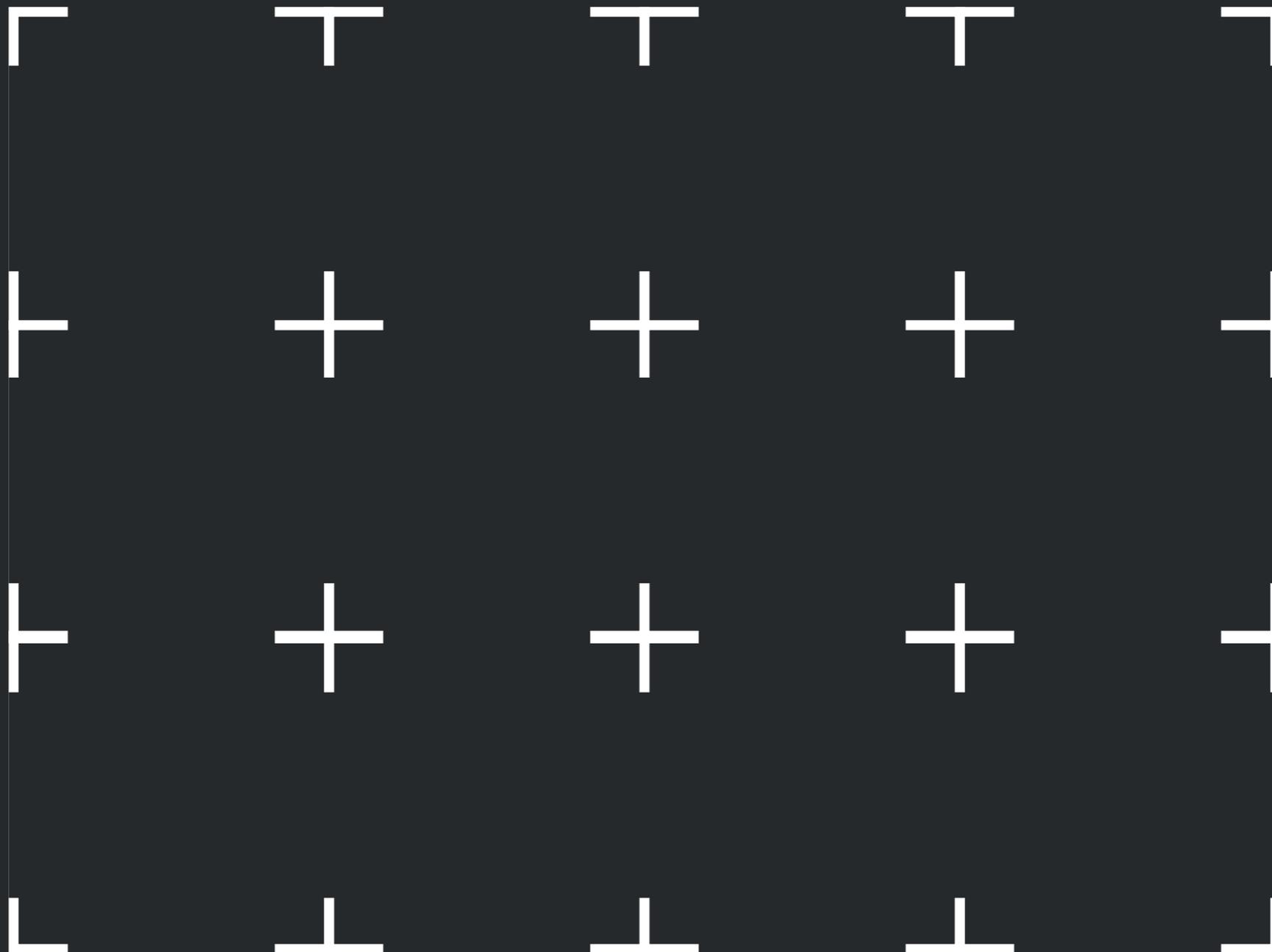
特殊な罫線

Gold Sponsors



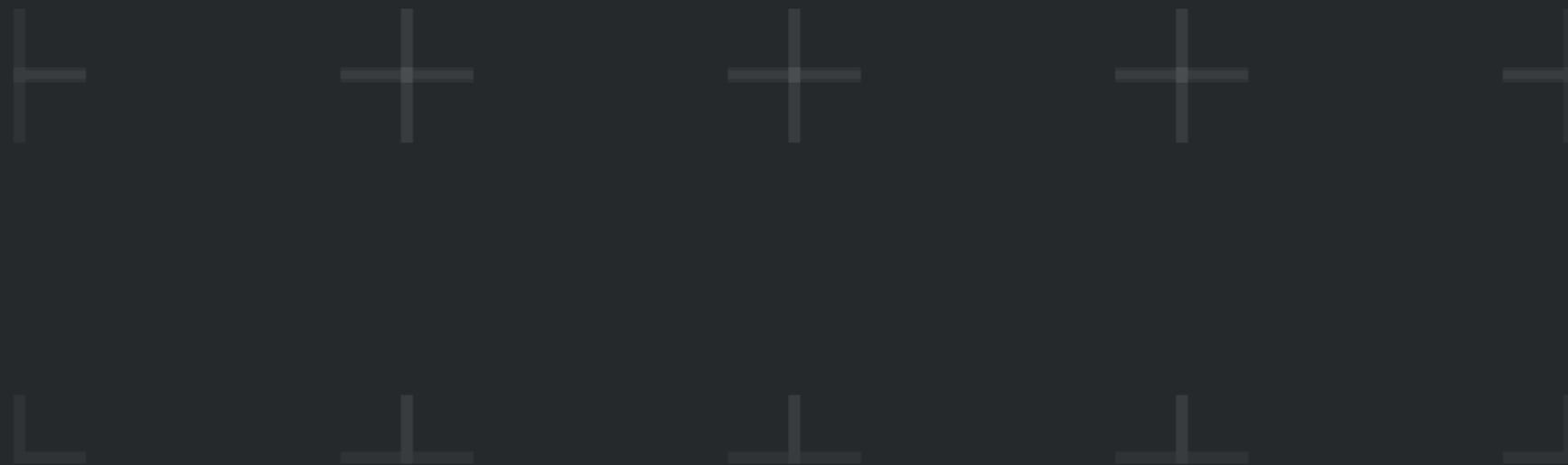
Silver Sponsors

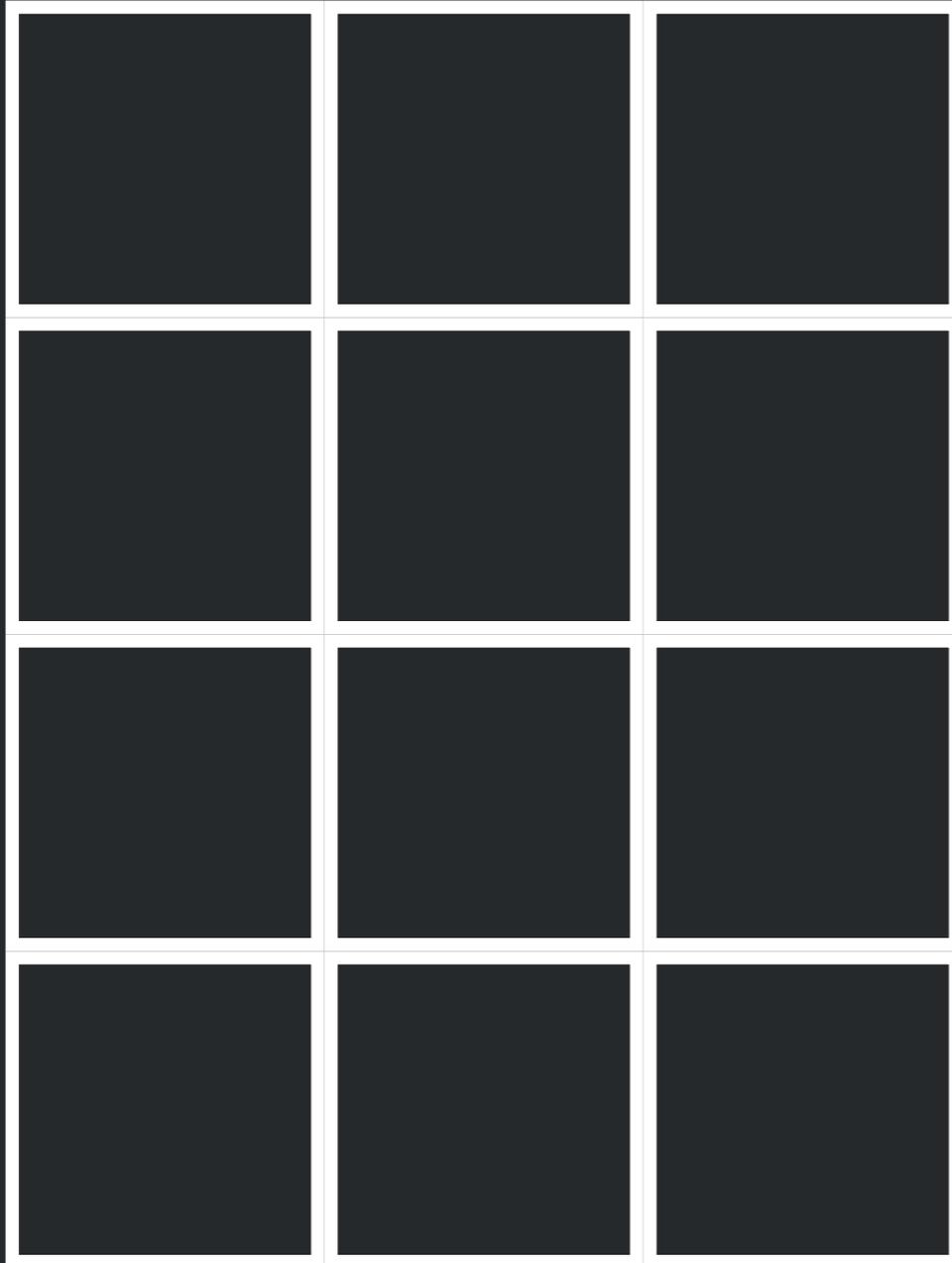




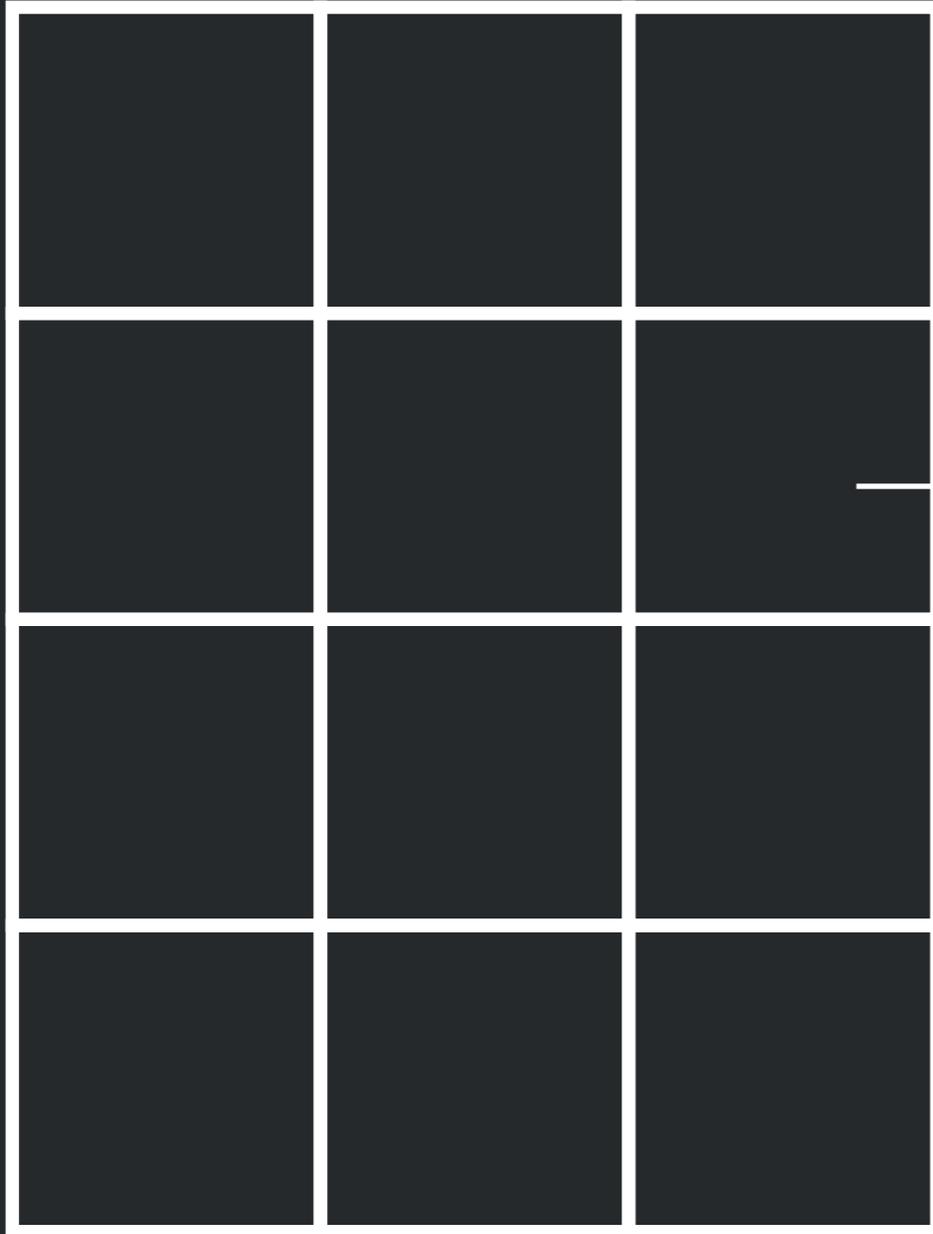


それぞれの罫線をこの形にできれば OK

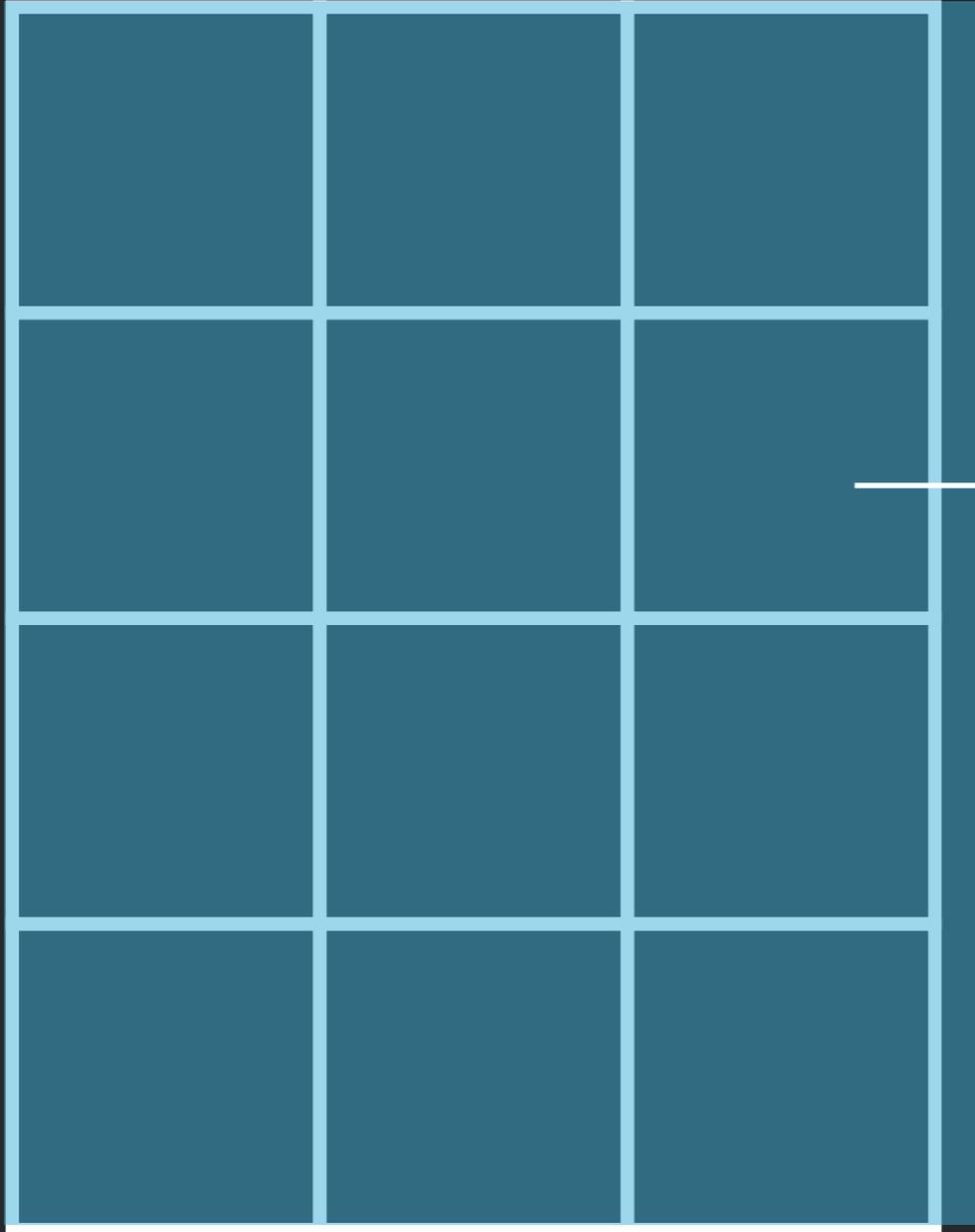




```
// まずは普通にボーダーをつける  
border: 1px solid;  
position: relative;  
z-index: 0; // 0 以下
```

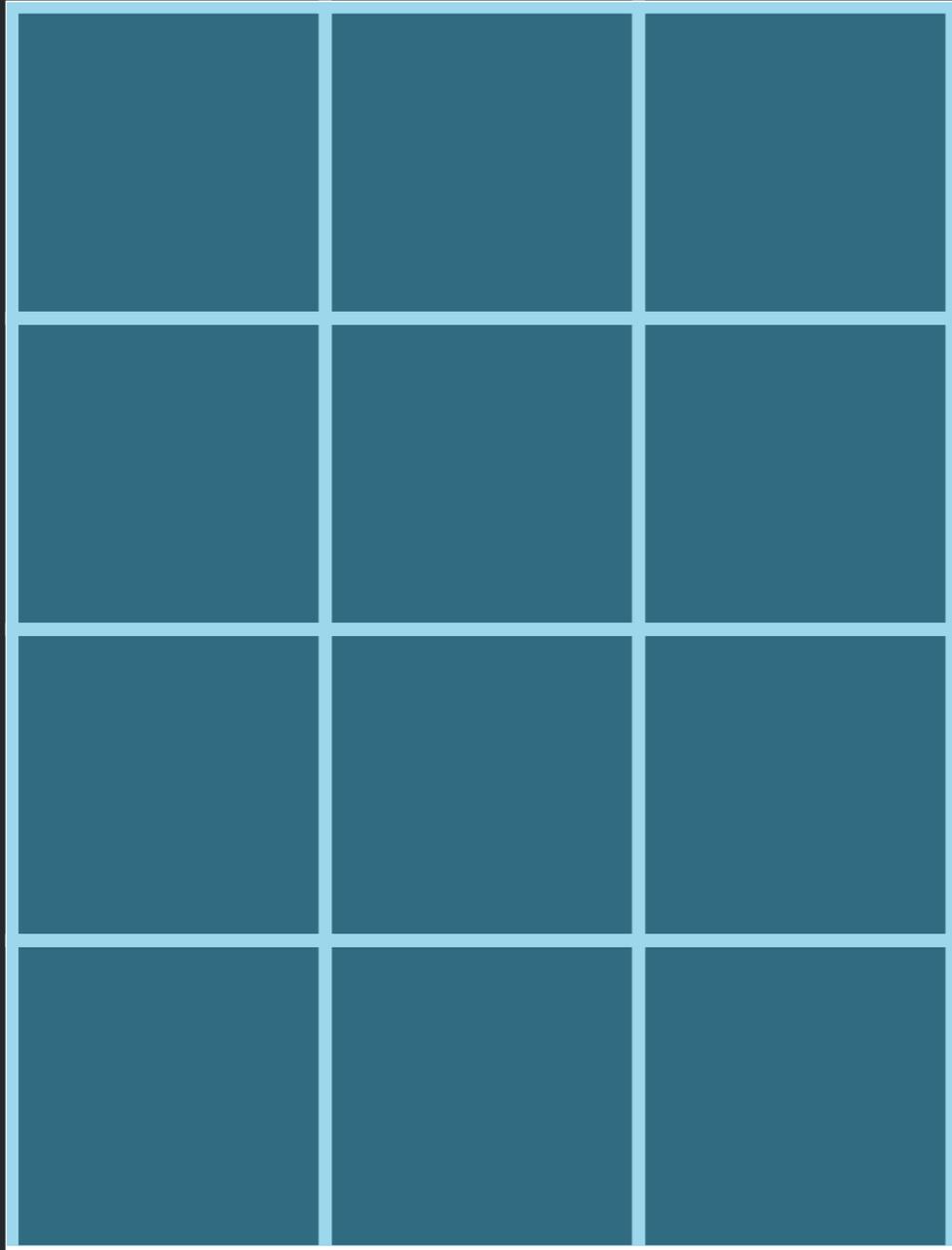


// ボーダーを重ねる
`margin: 0 -1px -1px 0;`



```
// ボーダーを重ねる  
margin: 0 -1px -1px 0;
```

重ねた分、コンテンツ幅が縮む

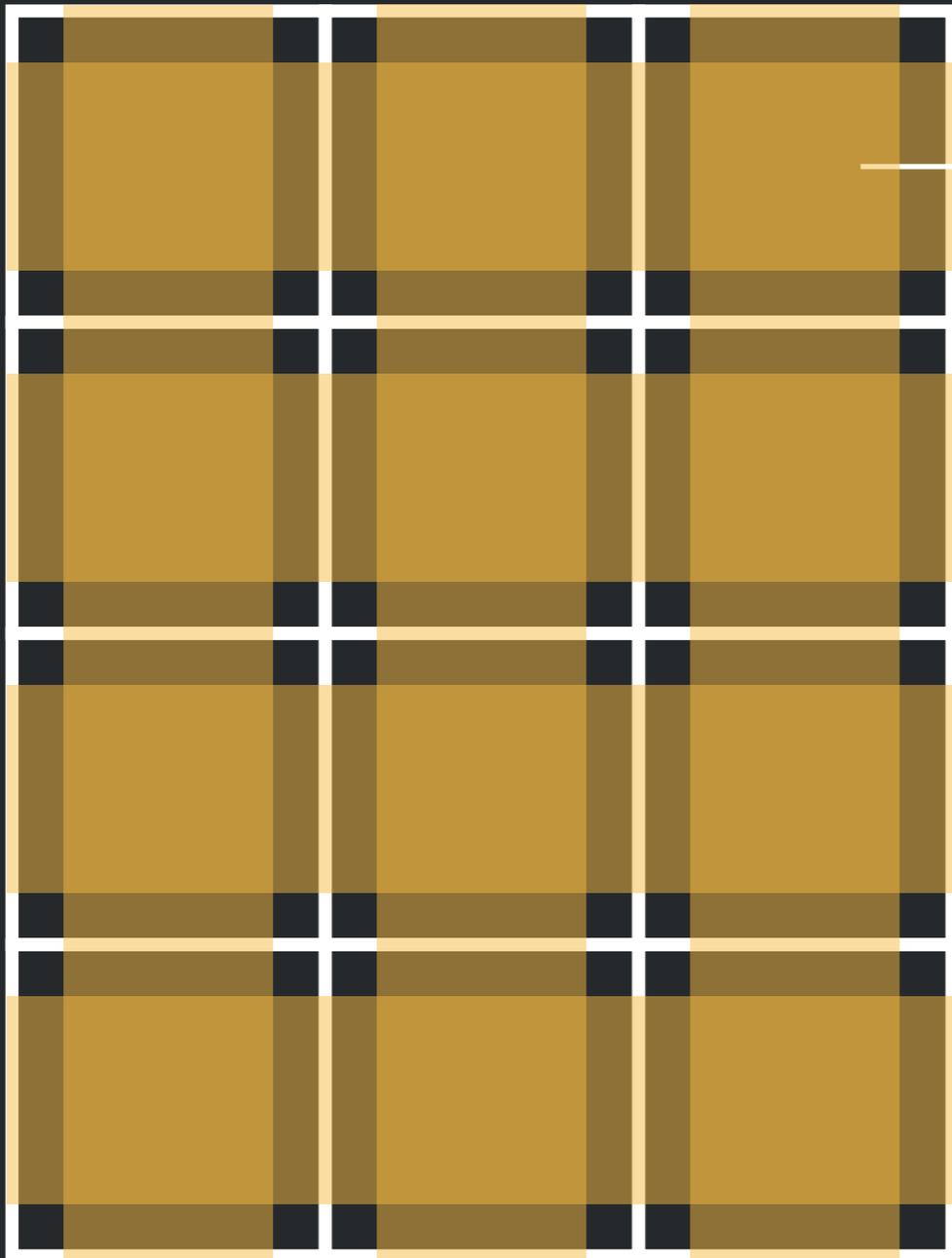


// 親要素に対して

// ボーダー幅×間のボーダー数分のネガティブマージン

```
margin-right: -2px;
```

これでピッタリ！



```
:before,  
:after {  
    // 背景色と同じ色を付ける  
    background: white;  
    content: "";  
    position: absolute;  
    z-index: -1; // -1 以下  
}  
  
:before {  
    top: 20px; // 十字の大きさ  
    bottom: 20px; // 十字の大きさ  
    left: -1px; // ボーダー幅  
    right: -1px; // ボーダー幅  
}  
  
:after {  
    top: -1px; // ボーダー幅  
    bottom: -1px; // ボーダー幅  
    left: 20px; // 十字の大きさ  
    right: 20px; // 十字の大きさ  
}
```

Demo: <https://ixkaito.github.io/unique-border/>

Source: <https://github.com/ixkaito/unique-border>

変態度：超ド級

特殊な飾り

NEWS

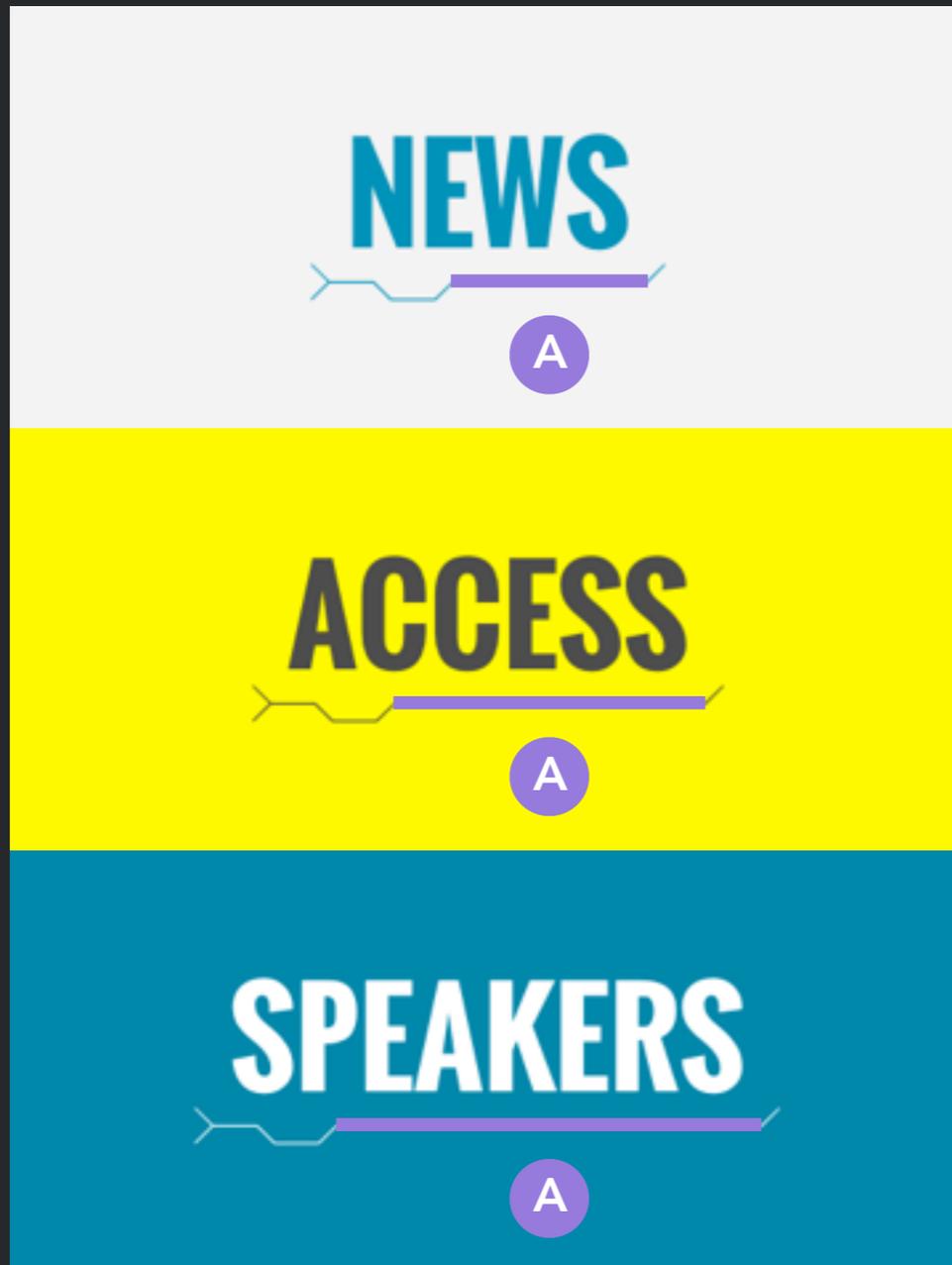


ACCESS

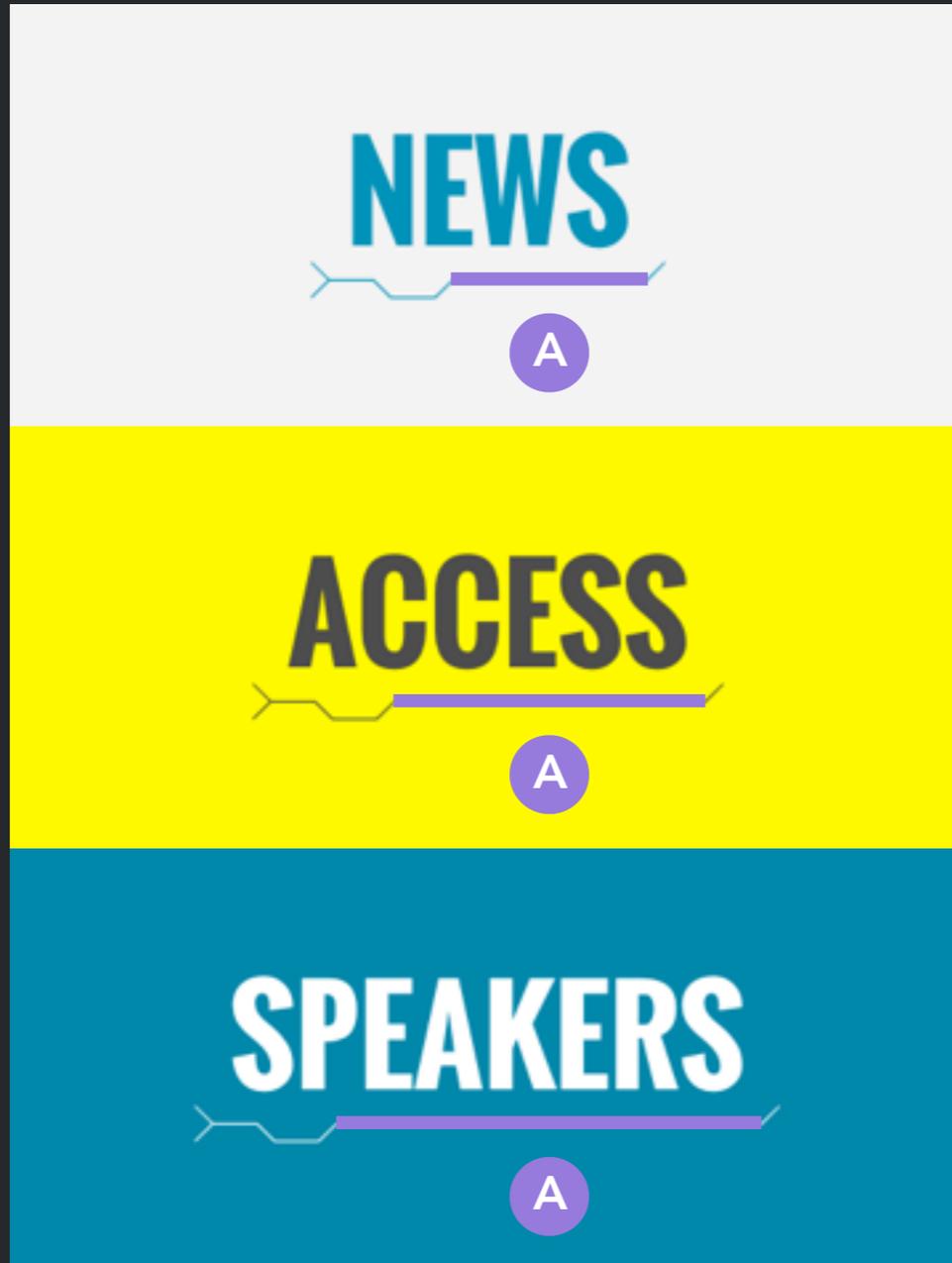


SPEAKERS





- テキストの長さによって A の部分が伸びる
- フォント色によって、飾りの色も変わる



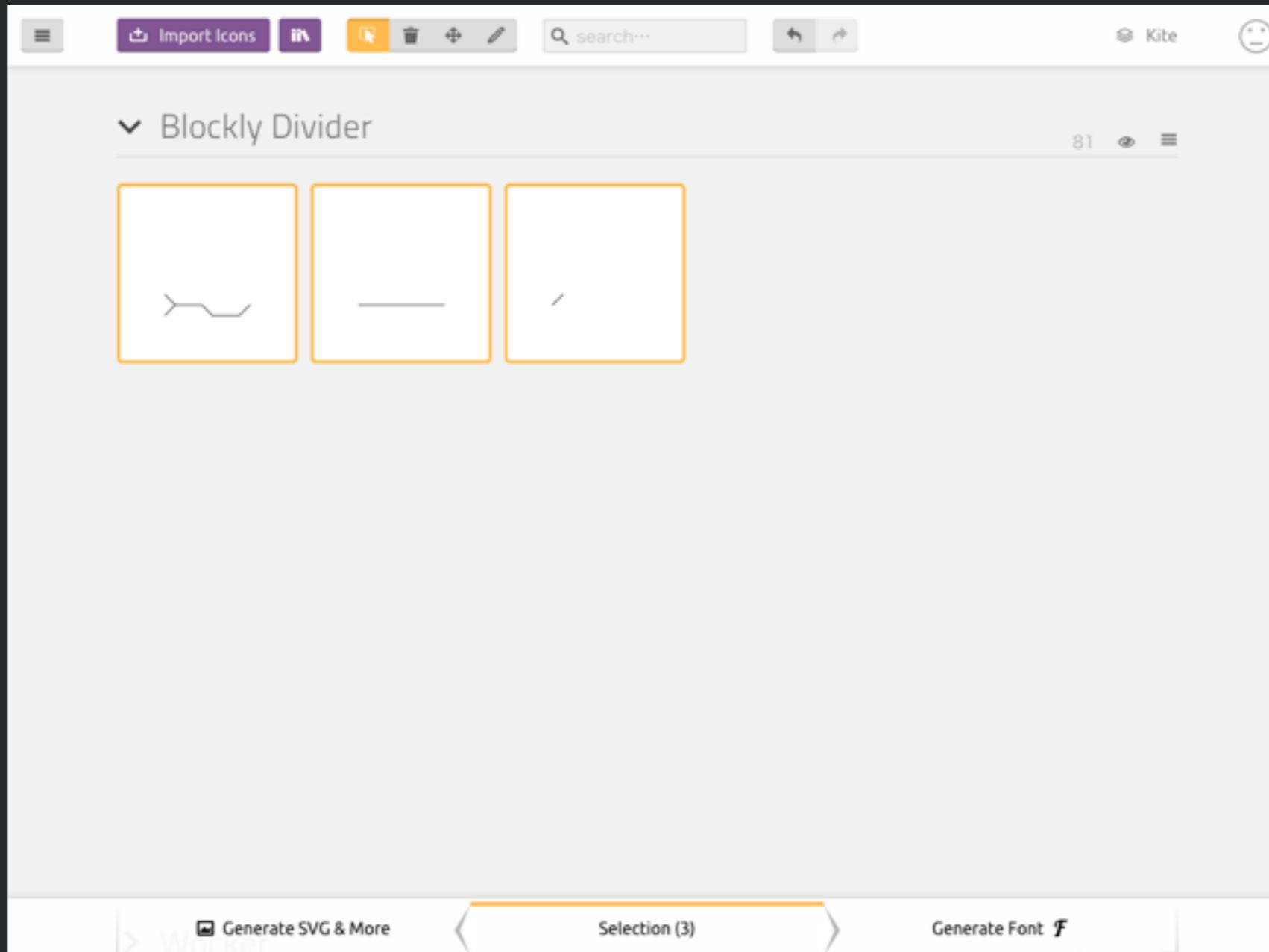
- テキストの長さによって A の部分が伸びる
先端、末端、中間部に分ける
- フォント色によって、飾りの色も変わる
フォントとして表示する

SVG を作る



- ピクセルレベルでぴったり合うように分割する
- アートボードは正方形の方が最終的にはキレイ
- 飾りがテキストの下の場合は、アートボードに対して下揃えがオススメ

IcoMoon でフォント化



HTML



SPEAKER

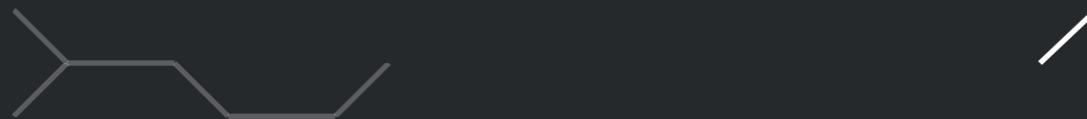
```
<h1 class="title">  
  <!-- インナーエレメントでラップする -->  
  <span class="title-inner">SPEAKER</span>  
</h1>
```

SPEAKER



`.title:before`

SPEAKER



`.title:after`

SPEAKER



```
.title-inner:after
```

SPEAKER



これを十分な長さにして、`overflow: hidden;` すればいいんじゃない？

SPEAKER



これを十分な長さにして、`overflow: hidden;` すればいいんじゃない？

どうやって？

SPEAKER



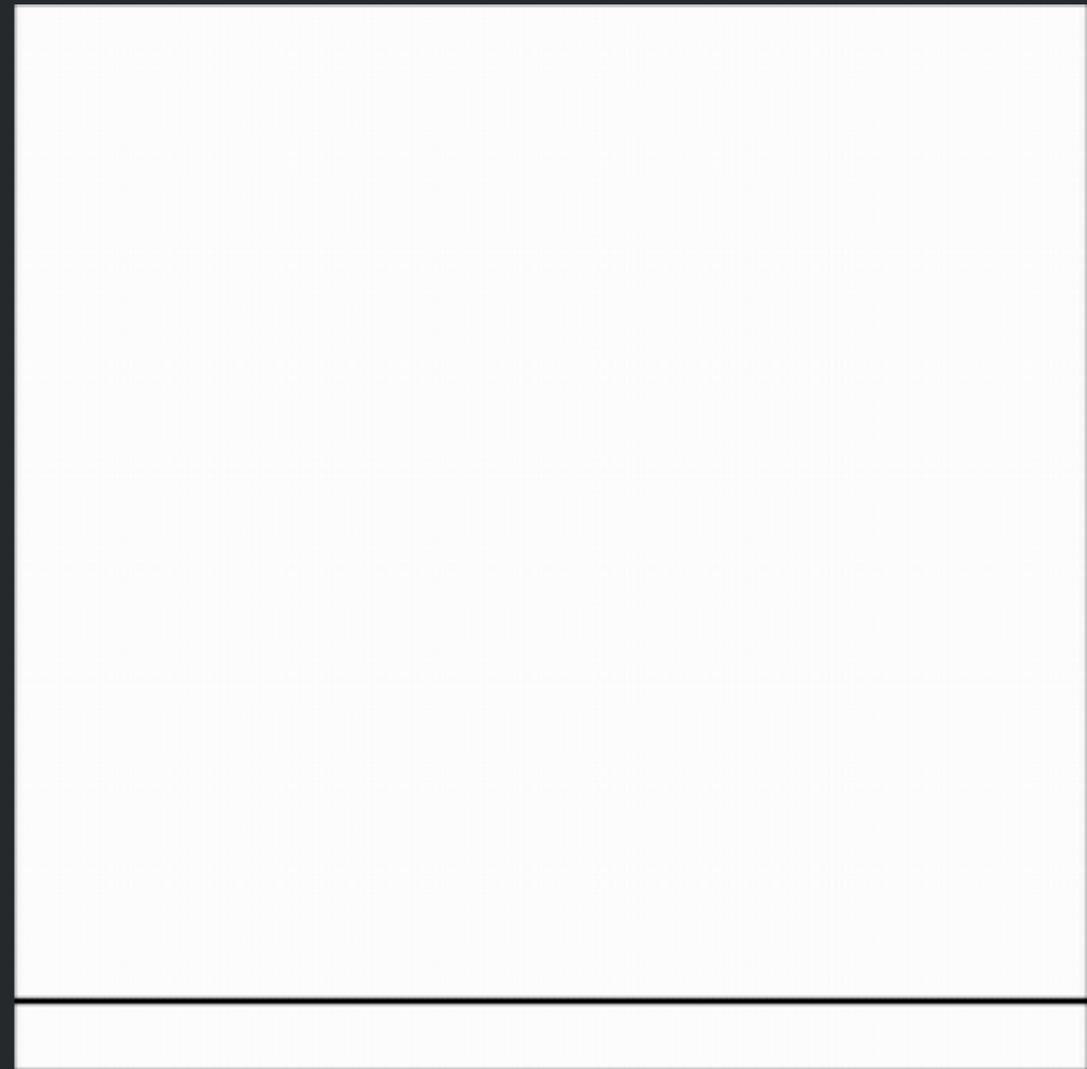
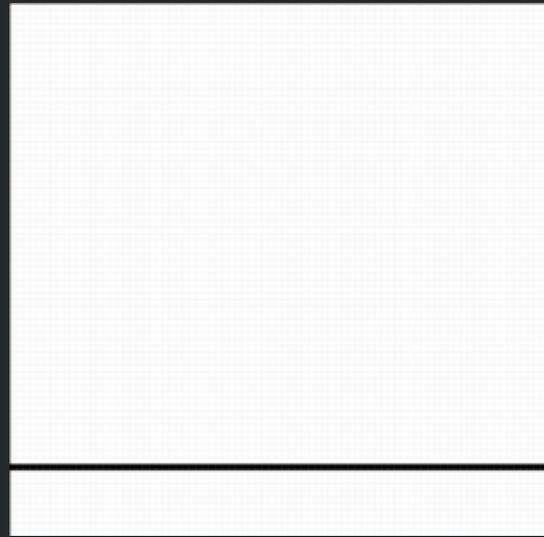
```
transform: scaleX(99);  
transform-origin: 0;
```

SPEAKER

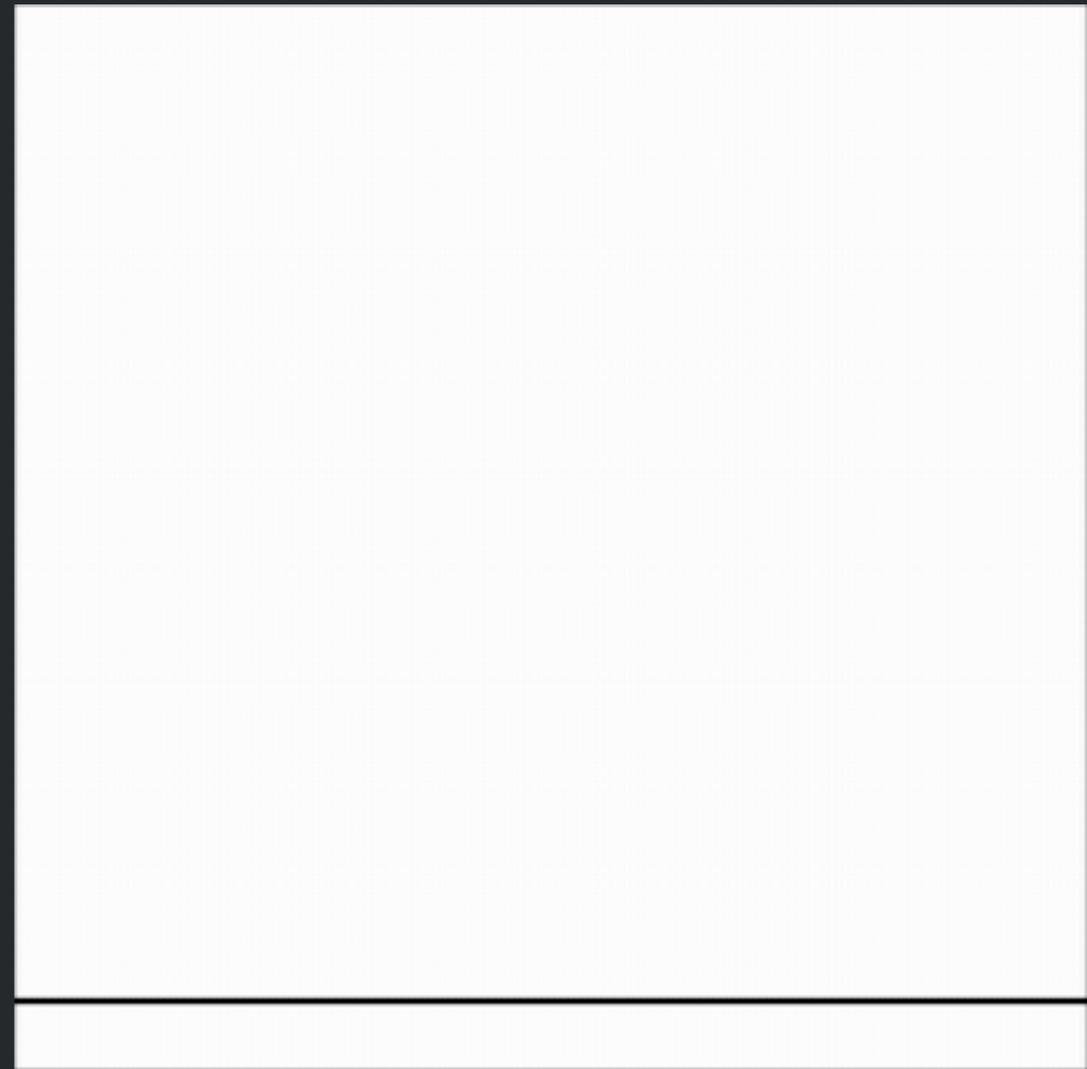
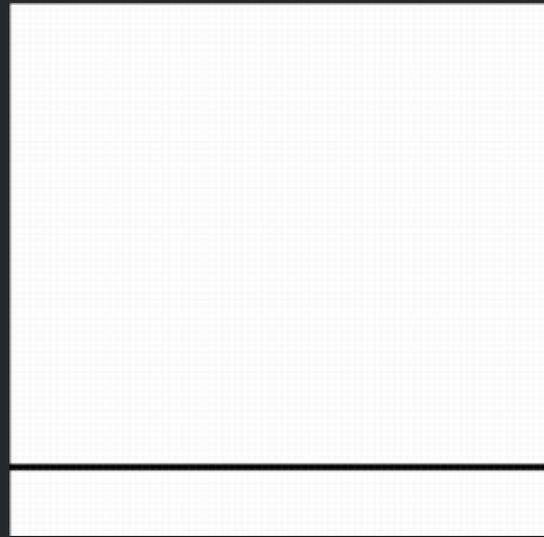


```
transform: scaleX(99);  
transform-origin: 0;
```

残念ながら、ブラウザによってズレる...



SVG のオリジナルサイズを
大きくする



SVG のオリジナルサイズを
大きくする

残念ながら、飾りの形が崩れる...

SPEAKER



```
content: "\e901\e901\e901\e901\e901\e901\e901\e901\e901\e901\e901\e901";
```

`content` に複数入れてつなげるのが一番キレイ

Demo: <https://ixkaito.github.io/unique-decoration/>

Source: <https://github.com/ixkaito/unique-decoration>

番外編

CSS4 :matches を使った CSS ハック

```
:-webkit-any( selector ) { /* Chrome */ }  
:-moz-any( selector ) { /* Firefox */ }  
:matches( selector ) { /* Safari */ }
```

UnresetCSS を作るため、
各ブラウザのデフォルトスタイルを調べてたら...

:matches とは、CSS4 の擬似クラスの一つで、
複数セレクターをまとめることができる。

```
:matches(article, aside, nav, section)
:matches(article, aside, nav, section) h1 {
  font-size: 1.17em;
}
```

これは...

```
article article h1,  
article aside h1,  
article nav h1,  
article section h1,  
aside article h1,  
aside aside h1,  
aside nav h1,  
aside section h1,  
nav article h1,  
nav aside h1,  
nav nav h1,  
nav section h1,  
section article h1,  
section aside h1,  
section nav h1,  
section section h1 {  
    font-size: 1.17em;  
}
```

これと同じ

```
article, aside, nav, section {
  article, aside, nav, section {
    h1 {
      font-size: 1.17em;
    }
  }
}
```

Sass などの入れ子に近い

各ブラウザの対応状況

```
:-webkit-any( selector ) { /* Chrome */ }
```

```
:-moz-any( selector ) { /* Firefox */ }
```

```
:matches( selector ) { /* Safari */ }
```

:matches は CSS3 では :any

この対応状況を逆手に利用して、
CSS のみでブラウザによってスタイルを変えることができる。

例えば、Safari だけ 1px ズレてボーダーが消えるのでボーダーを 2px にする。(実例)

まとめ

- これらは変態的 CSS トリックのほんの一部
- ここで取り上げたやり方を覚える必要はない、重要なのは下記
- CSS でもプログラミング的思考力や創造性、問題解決力、男のロマン (この場合女性でも男のロマン) は必要
- 1ピクセルへのこだわりはデザインの再現性やサイトのクオリティを格段に引き上げる

最後に

KITERETZ inc. (株式会社キテレッツ) では、
魔法使いになりたい人を募集します。
リモート、パートタイム歓迎。

サイトから:

<http://kiteretz.com/>

Facebook メッセージから:

<http://facebook.com/kite.koga>

GitHub Issue:

<https://github.com/kiteretz>