

CSSNITE

CSS Nite

KOBE

Vol.34

2018.5.12 

／ CSS Nite in KOBE ／

CSSの最新テクニックを使って
一歩先取るコーディング

Presented by Mana with 

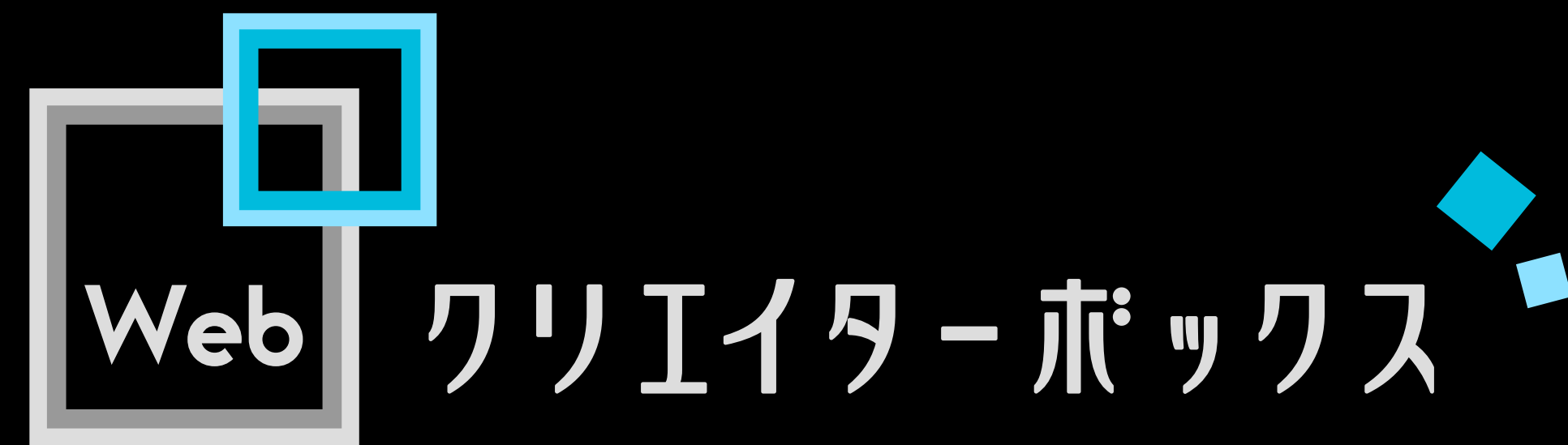
Mana



広島弁は

しゃべれないじゃけん

ブログ「Webクリエイターボックス」



webcreatorbox.com

カナダ、イギリス、オーストラリアで就労



「Tech Academy」でインストラクター



techacademy.jp







CSS Gridを使ったレイアウト

これまでのCSSレイアウト組み

table



これまでのCSSレイアウト組み

table

float



これまでのCSSレイアウト組み

table

float

flex



これまでのCSSレイアウト組み

table

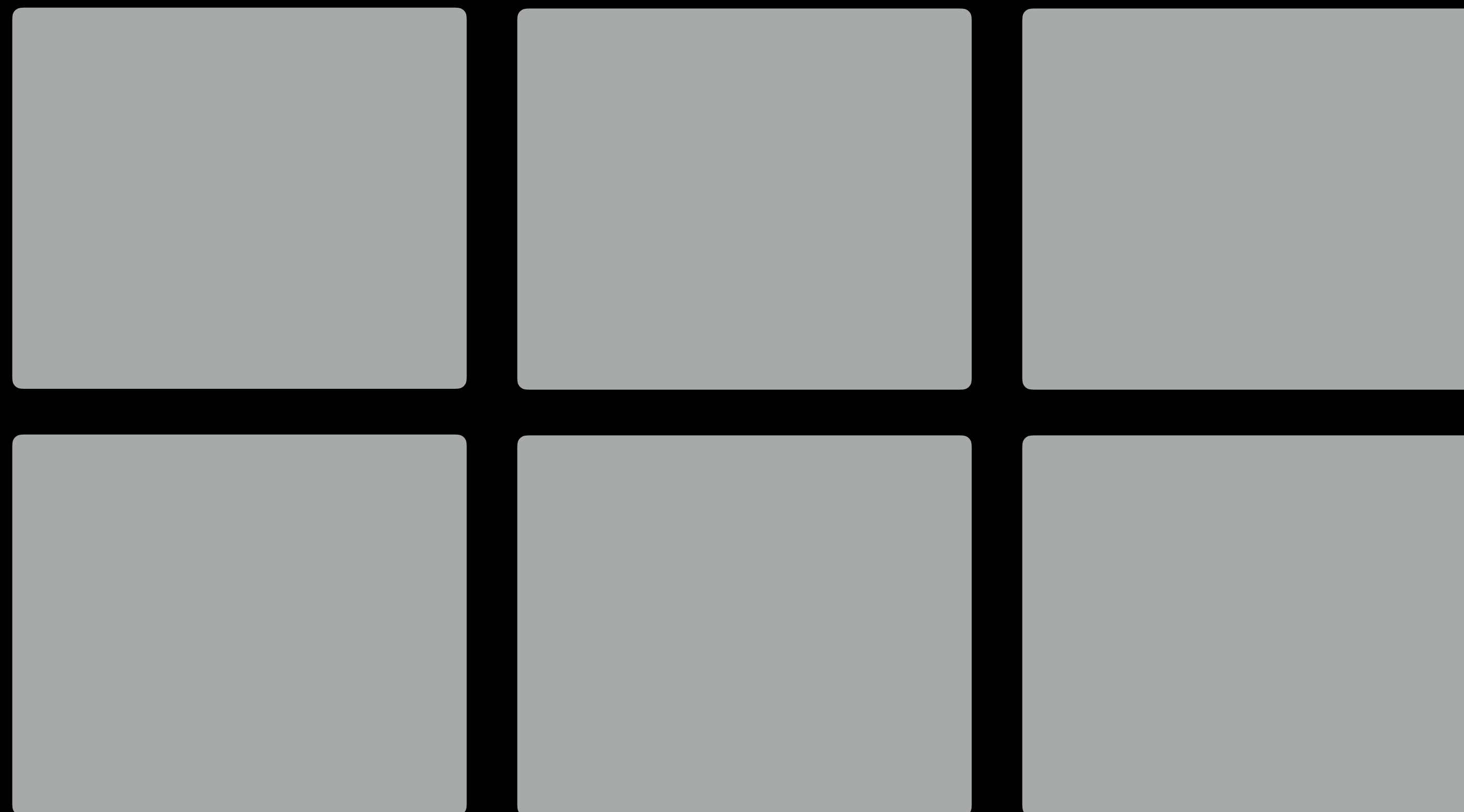
float

flex

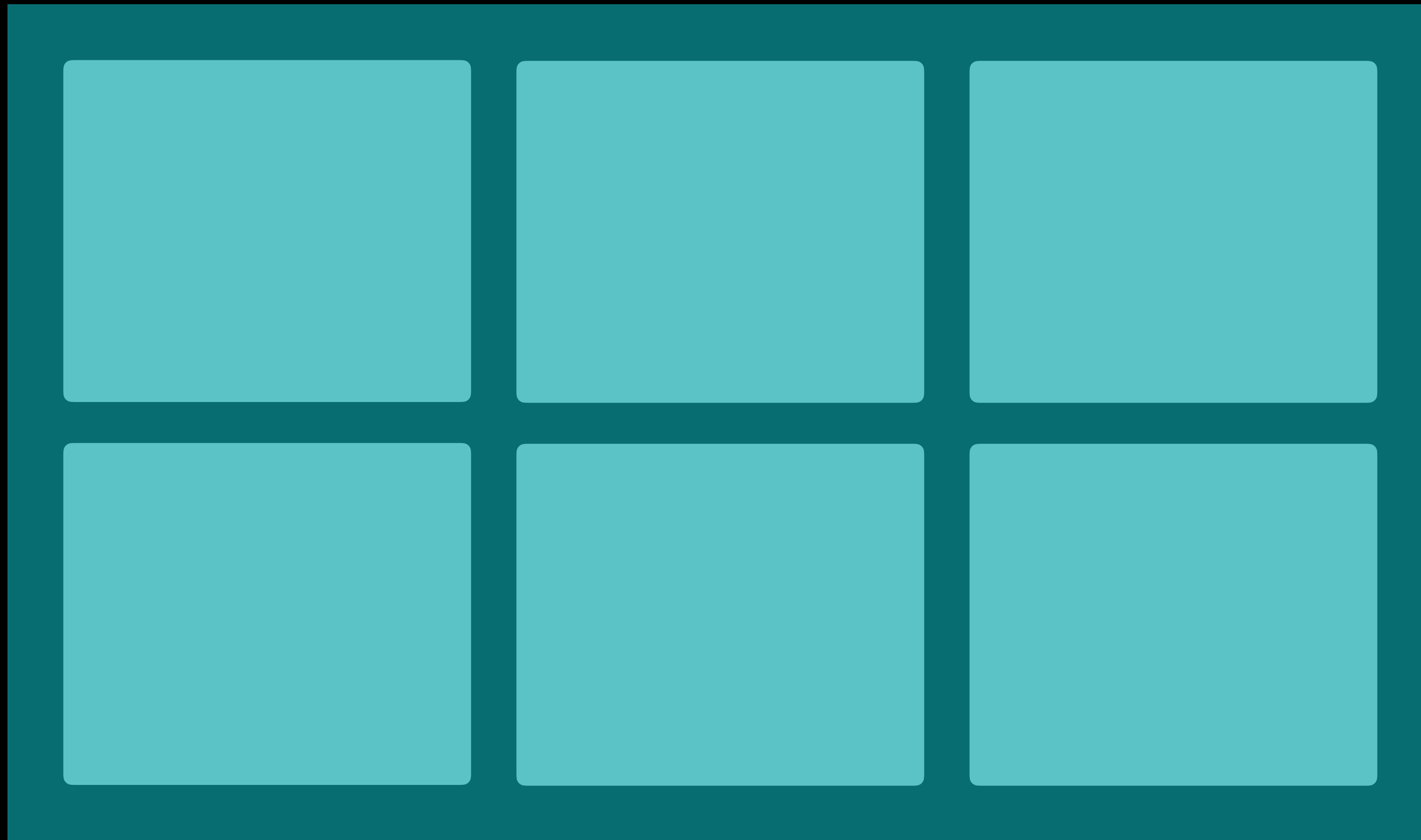
grid 



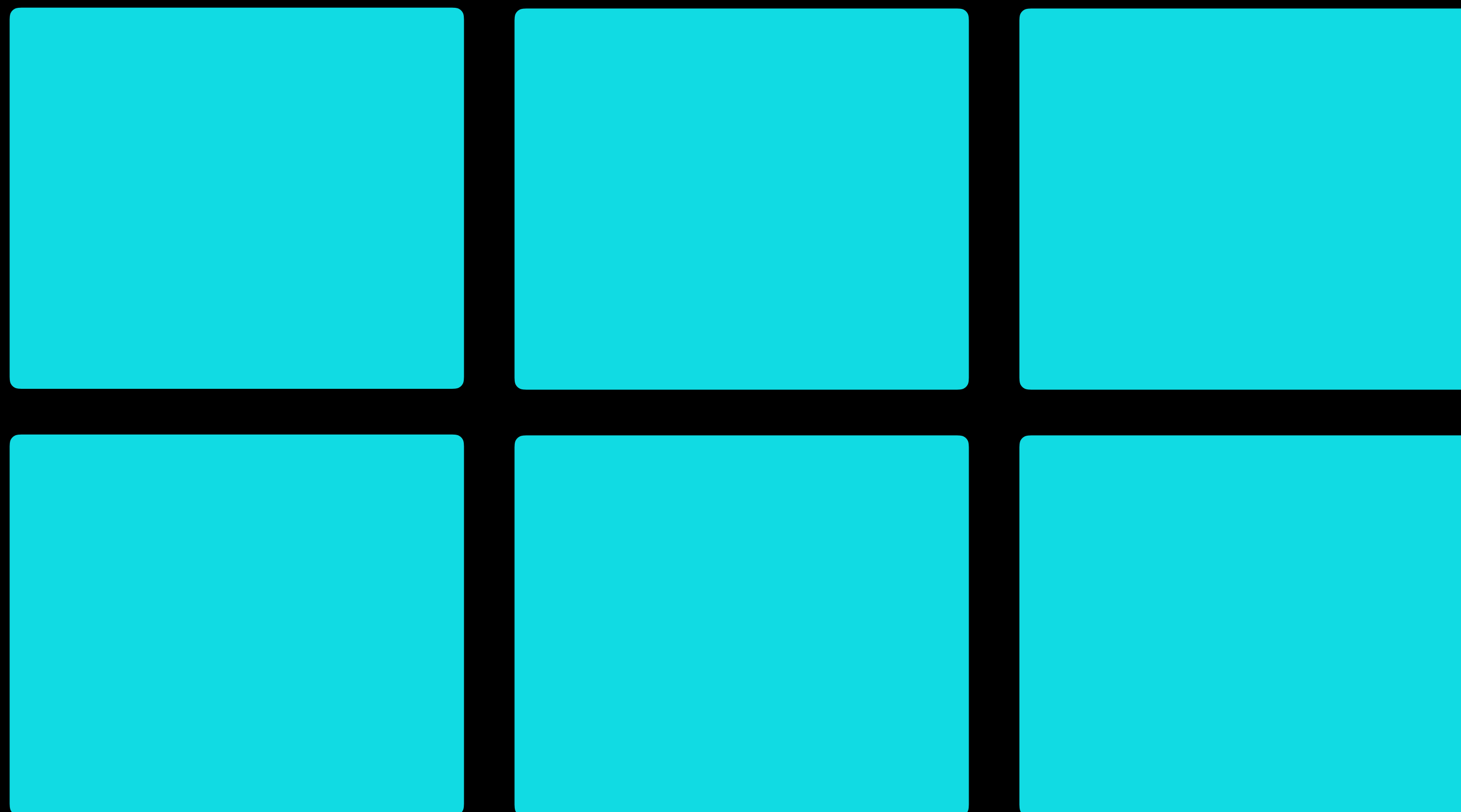
CSS Gridの構成



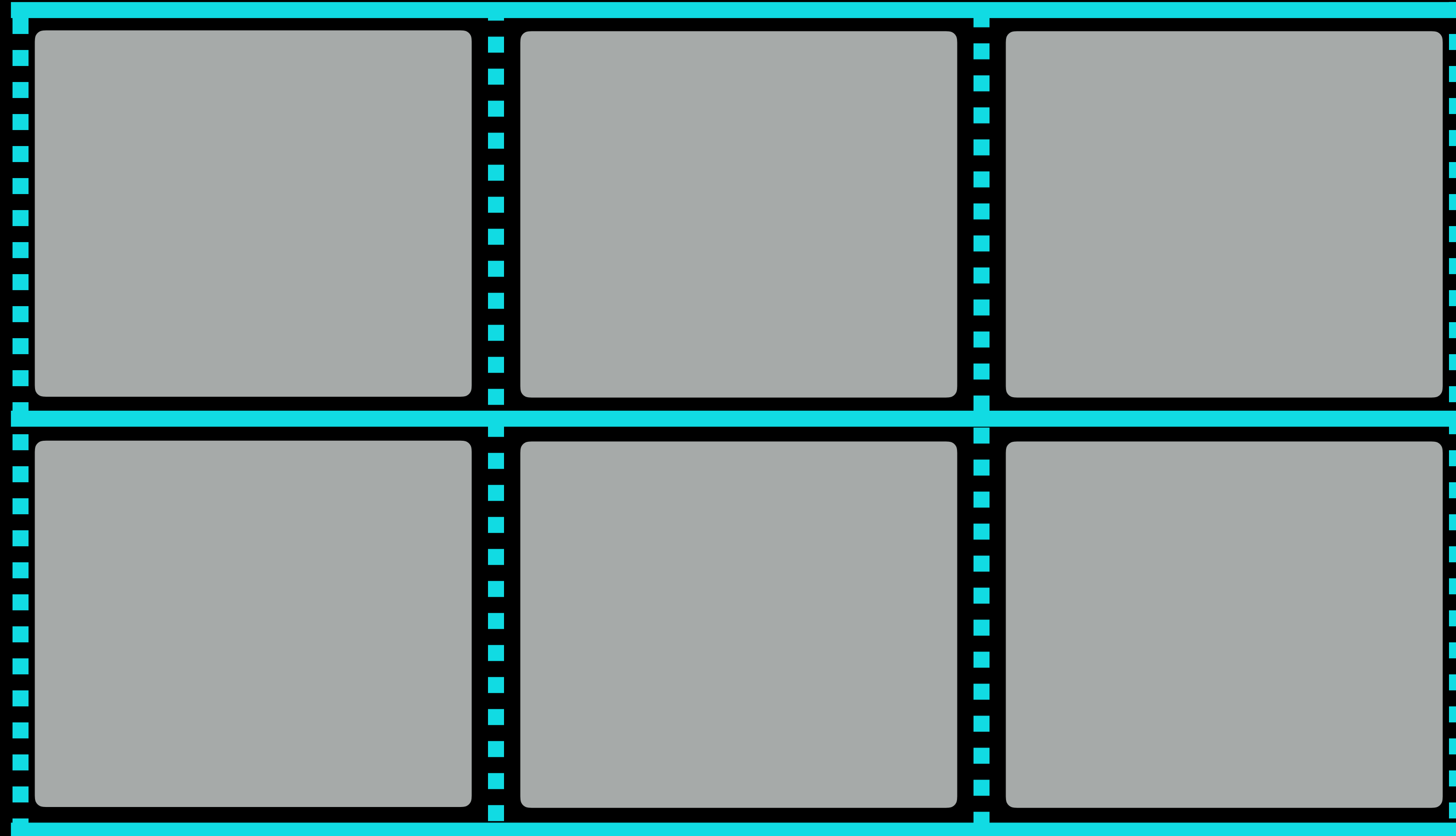
グリッドコンテナ（親要素）



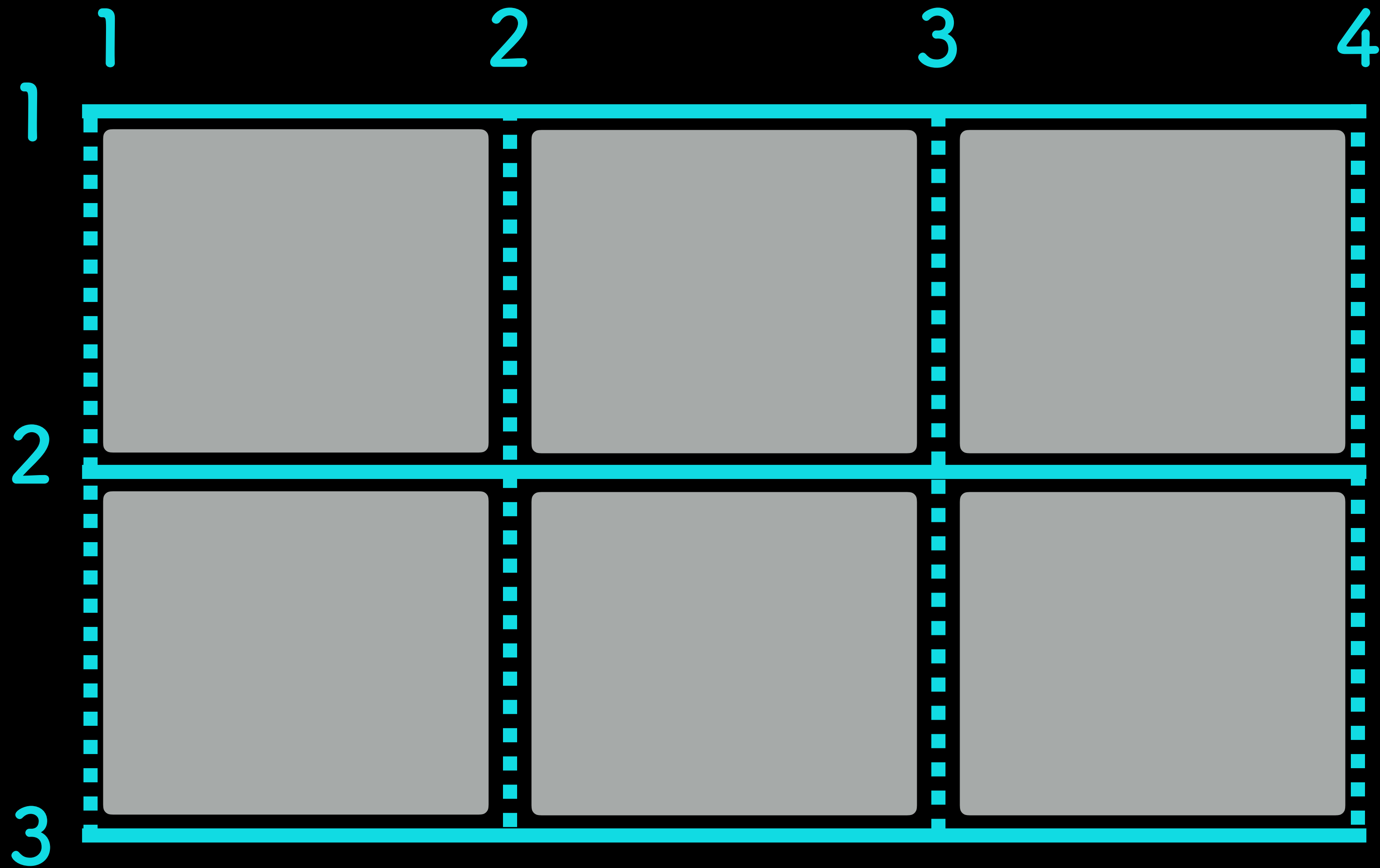
グリッドアイテム (子要素)



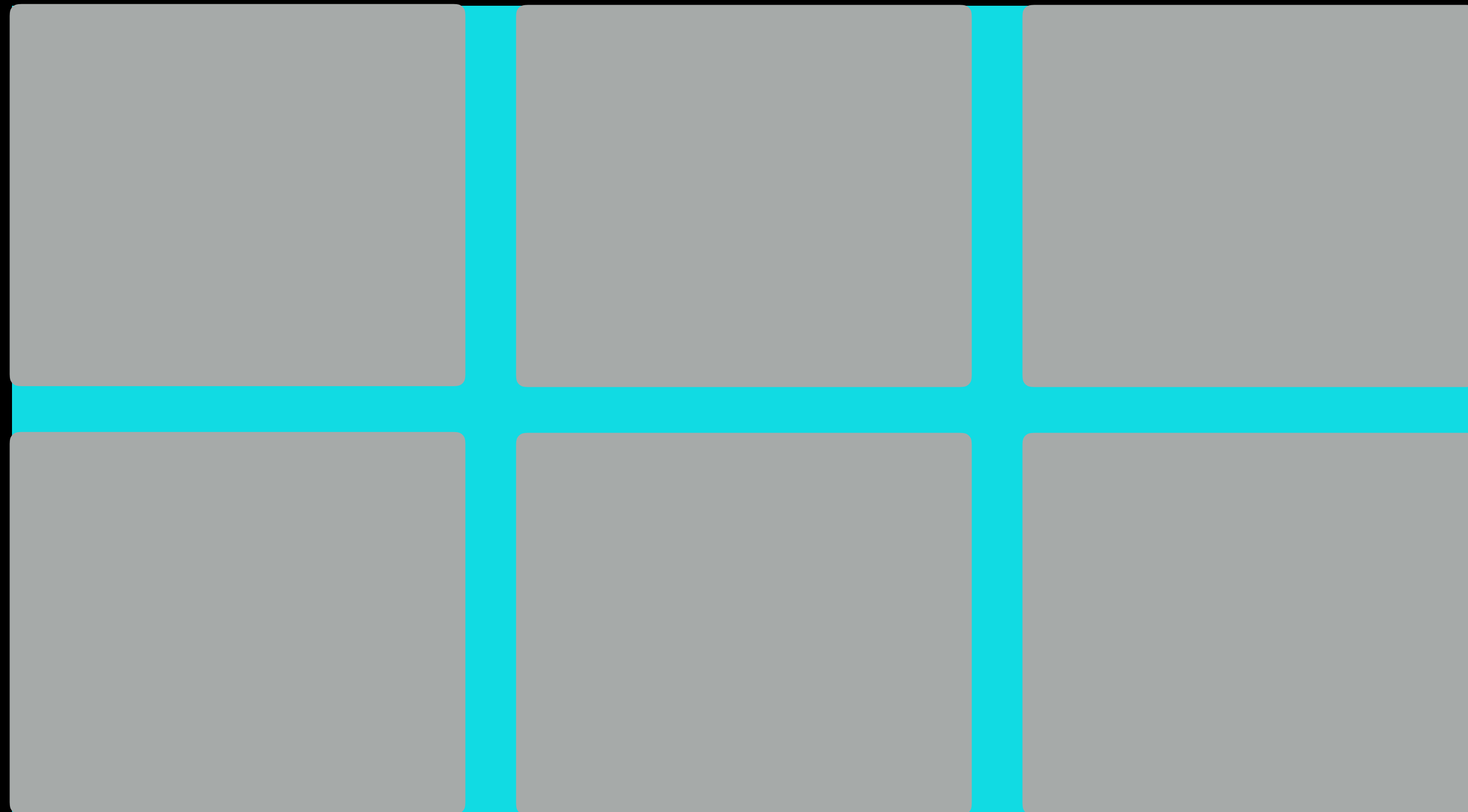
グリッドライン



グリッドライン

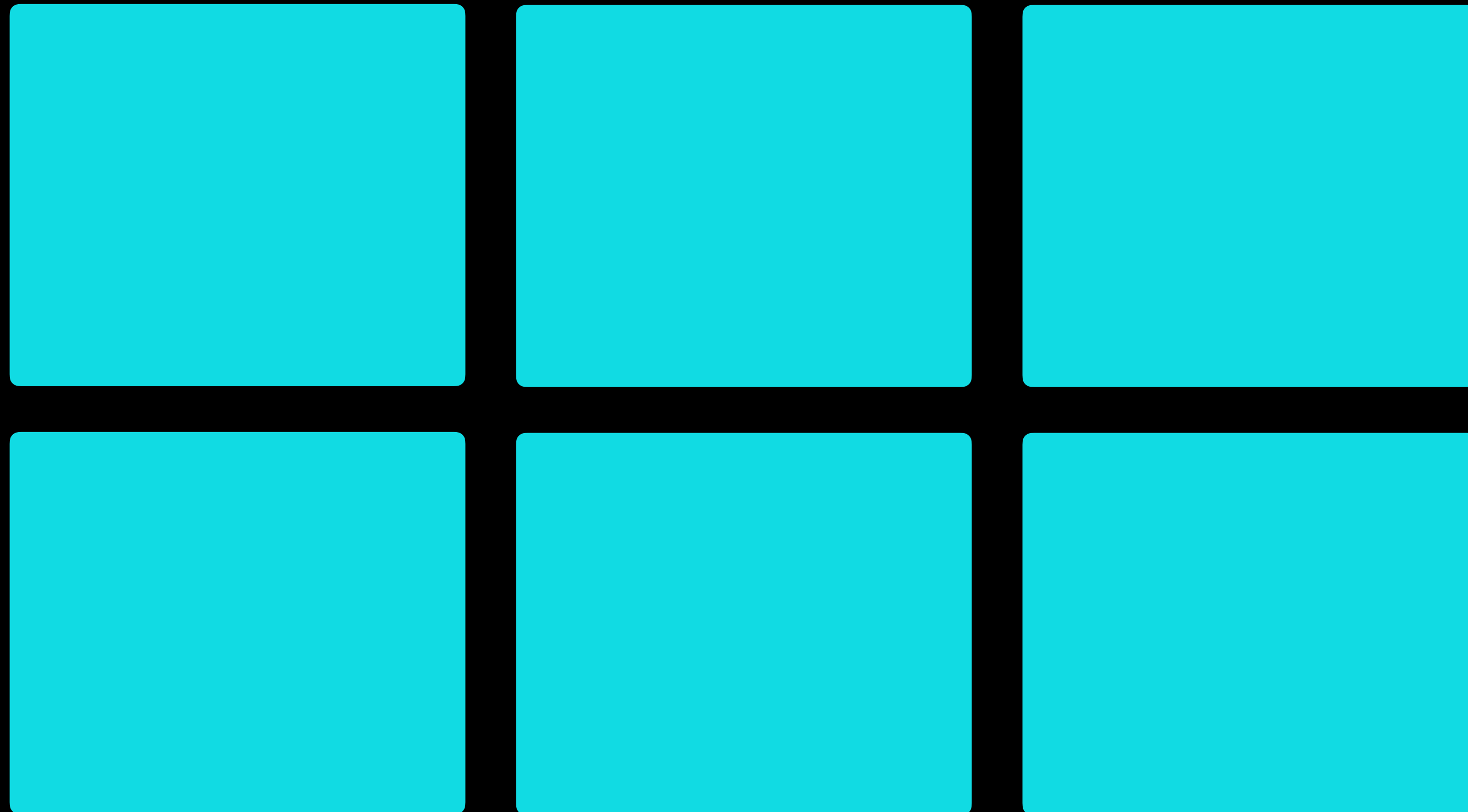


グリッドギャップ



CSS Gridを使ってみよう!

基本の使い方

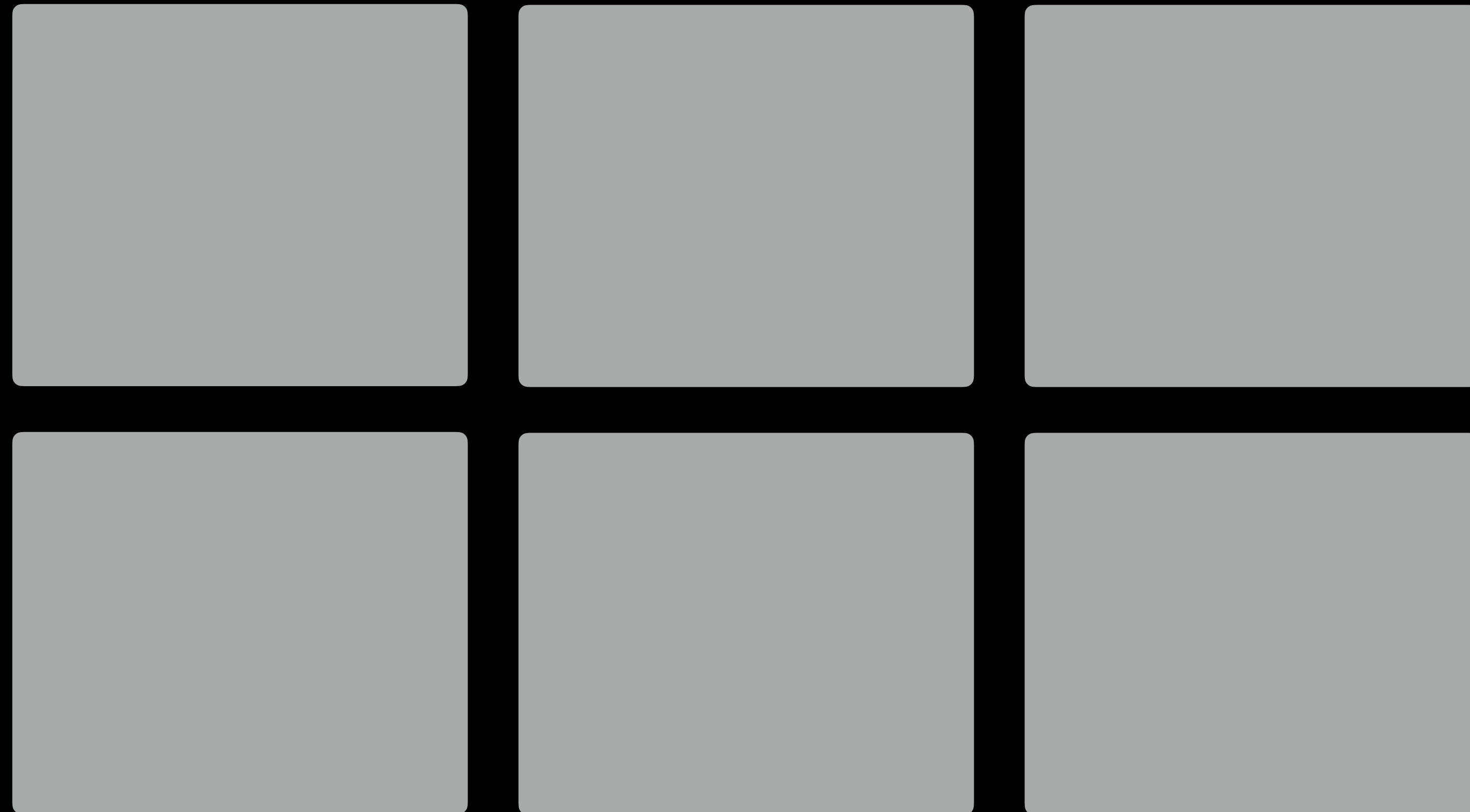


基本の使い方

- 
- `grid-template-columns`
 - `grid-template-rows`
 - `grid-gap`

Demo

サイズの異なるグリッドアイテムを並べる



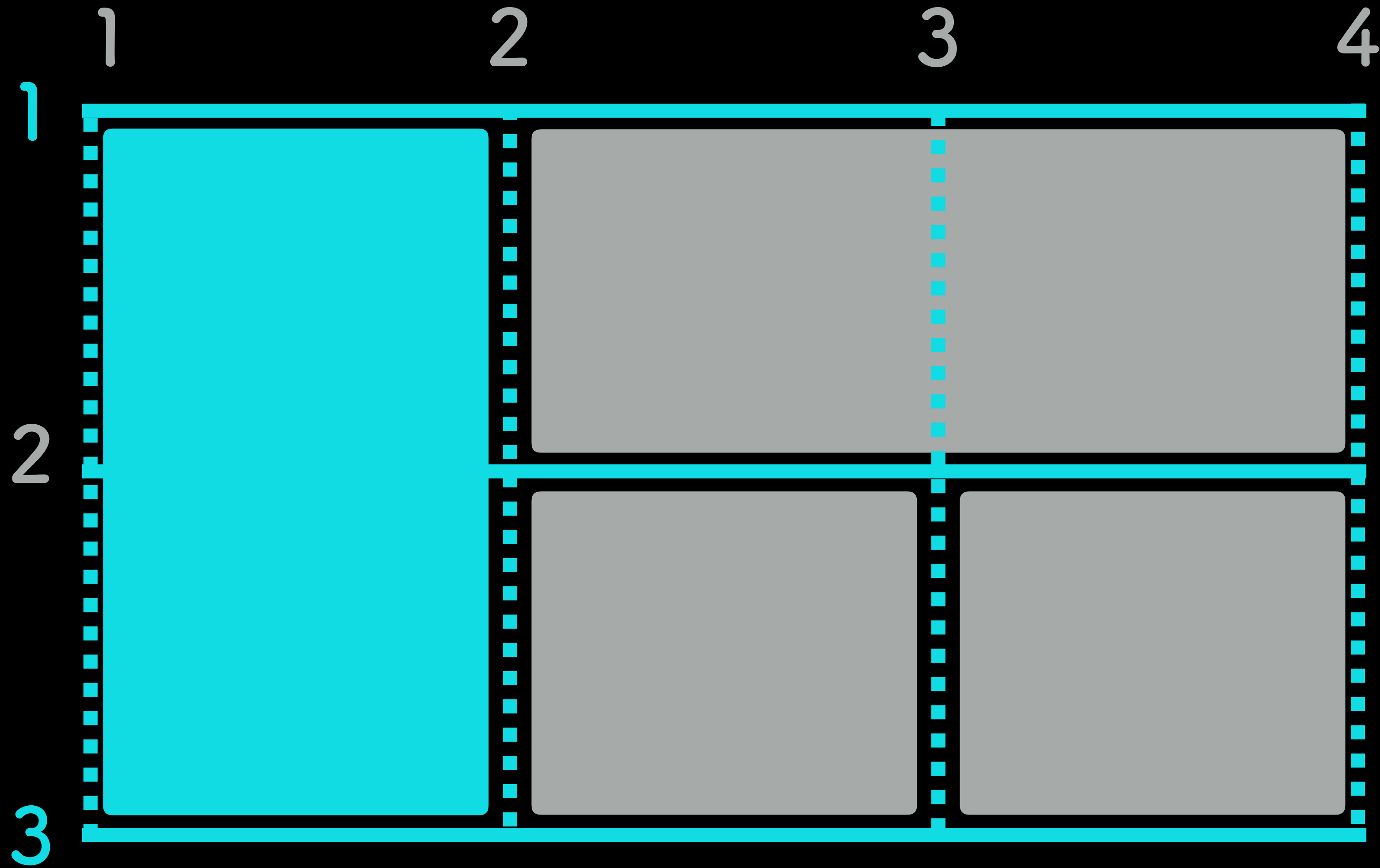
サイズの異なるグリッドアイテムを並べる



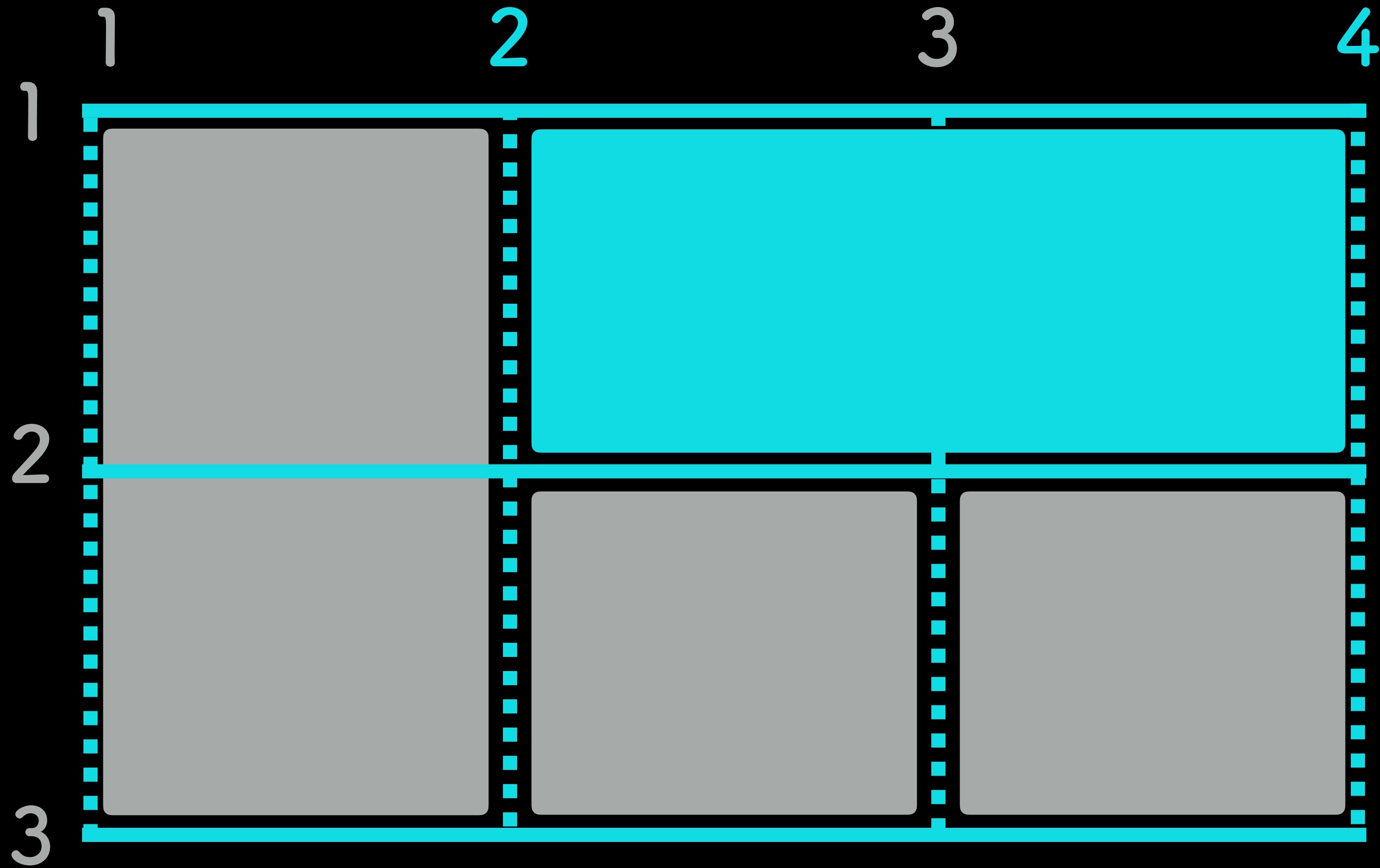
サイズの異なるグリッドアイテムを並べる

- grid-column
- grid-row

grid-row: 1 / 3;



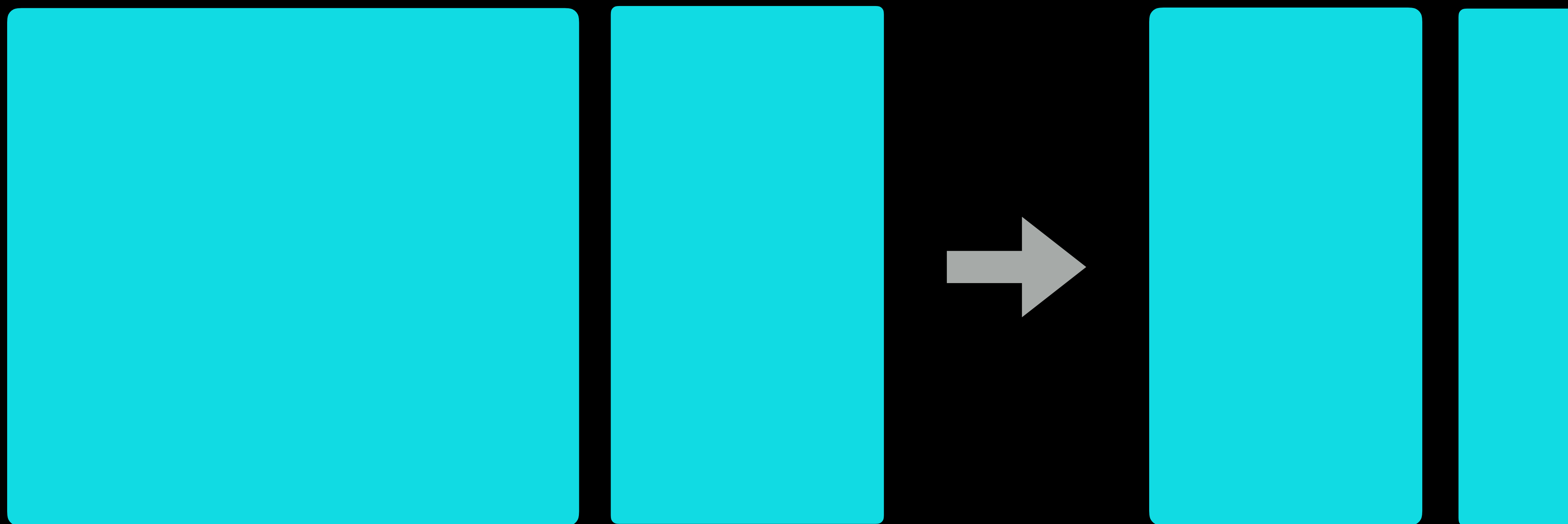
grid-column: 2 / 4;



Demo

伸縮する2カラム

伸縮する2カラム



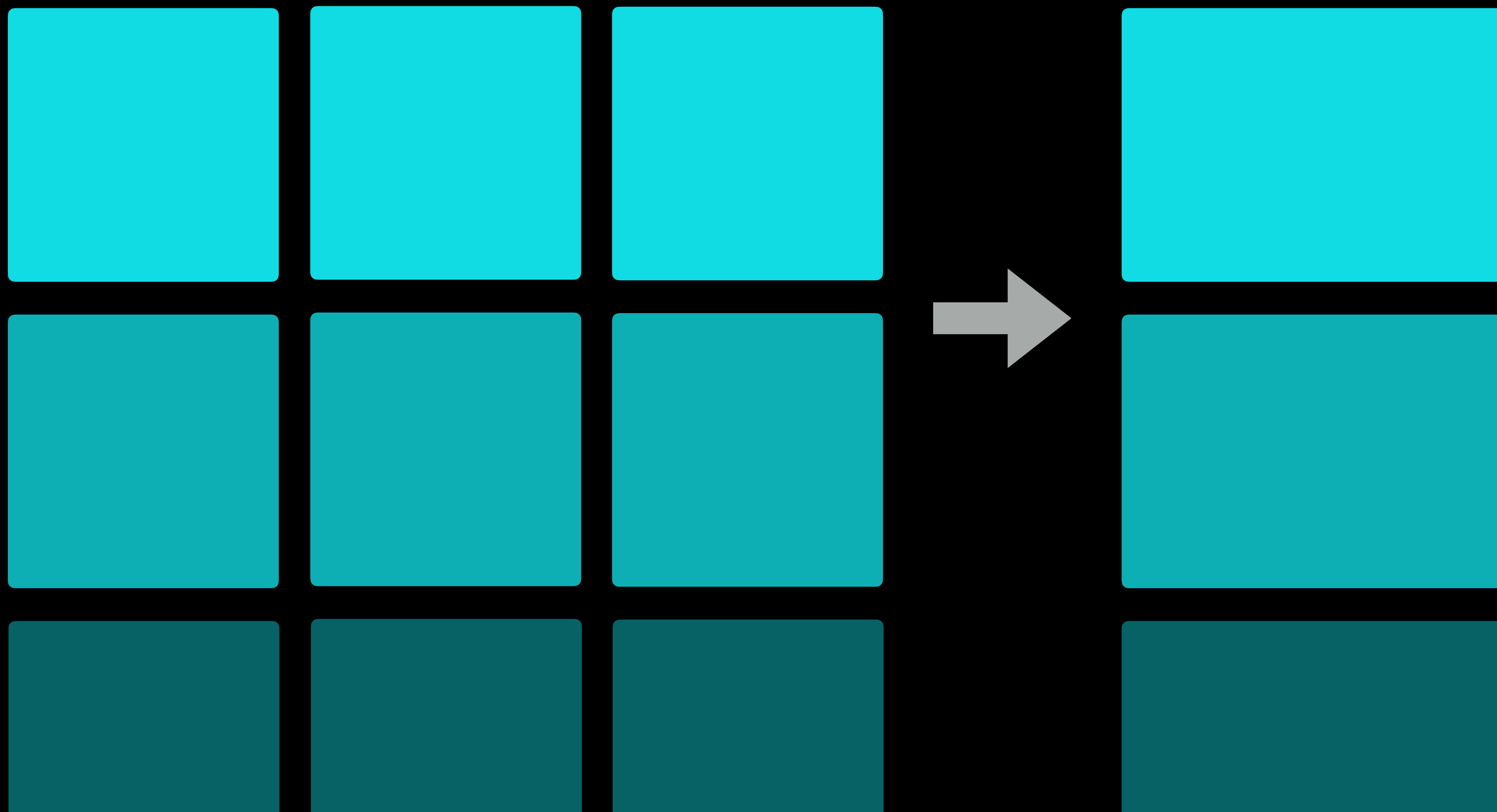
伸縮する2カラム

- fr (単位)
- minmax

Demo

伸縮するマルチカラム

伸縮するマルチカラム



伸縮するマルチカラム

- repeat
- auto-fit

Demo

対応ブラウザ

Chrome	Safari	Firefox	Edge	IE
✓ 57+	✓ 10.1+	✓ 52+	✓ ※ 12+	✓ ※ 10+

※ -ms- ベンダープレフィックスが必要

CSS 変数 で管理しやすくする

CSS 変数 とは ?

CSS変数

- CSS custom properties for cascading variables

CSS変数

- CSS custom properties for cascading variables
- カスケード変数のためのCSSカスタムプロパティ

CSS変数

- CSS custom properties for cascading variables
- カスケード変数のためのCSSカスタムプロパティ
- CSSカスタムプロパティ

CSS変数

- CSS custom properties for cascading variables
- カスケード変数のためのCSSカスタムプロパティ
- CSSカスタムプロパティ
- CSS変数 ←

変数???

$$X = 10$$

$$Y = 20$$

$$X + Y = 30$$

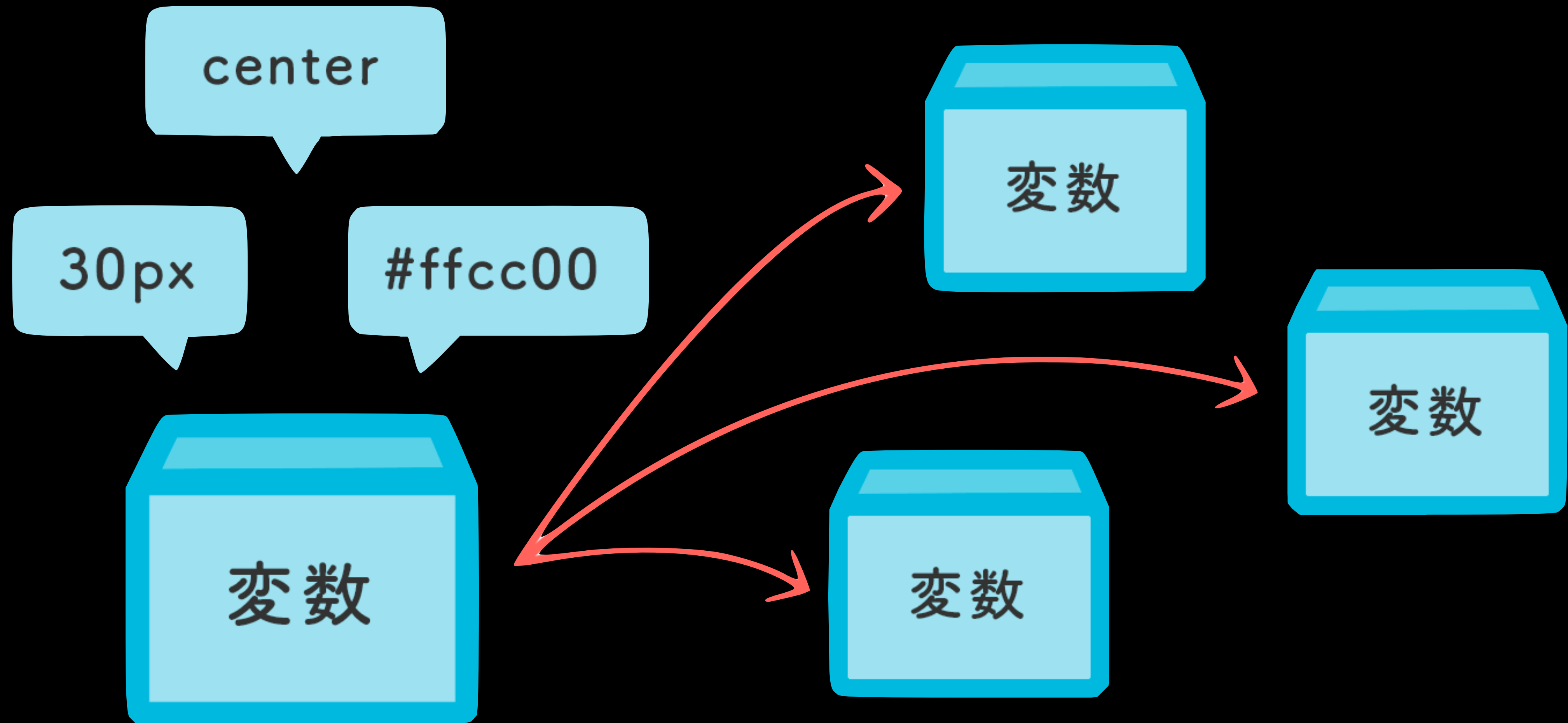
$$X = 10$$

$$Y = 20$$

$$X + Y = 30$$

これが
変数

何度でも使える!



CSS変数の使い方

1. CSS変数の定義

2. 定義したCSS変数を呼び出す

1. CSS変数の定義

```
--main-gutter: 12px;
```

変数名

値

1. CSS変数の定義

```
:root {  
  --main-gutter: 12px;  
  --main-position: center;  
  --main-lh: 1.5;  
}
```

2. 定義したCSS変数を呼び出す

```
margin: var(--main-gutter);
```

変数名

Demo

様々な場面で

CSS変数を使ってみよう

＼ 様々な場面でCSS変数を使ってみよう ① ／

変数の中で変数を使う

変数の中で変数を使う

```
:root {  
  --gutter: 10;  
  --base-gutter: calc(var(--gutter) * 3px);  
}
```

変数の中で呼び出す

Demo

＼ 様々な場面でCSS変数を使ってみよう ② ／

メディアクエリーとCSS変数

Demo

＼ 様々な場面でCSS変数を使ってみよう ③ ／

CSS変数が取得できなかった時の
フォールバック

CSS変数が取得できなかった時のフォールバック

```
.box {  
  color: var(--my-color, #f66);  
}
```

カンマで区切る

CSS変数を使う時の**注意点**

＼ CSS変数を使う時の注意点 ① ／

大文字と小文字が区別される

大文字と小文字が区別される

```
:root {  
  --bg: pink;  
  --BG: orange;  
}
```


＼ CSS変数を使う時の注意点 ② ／

プロパティ名を変数にできない

プロパティ名を変数にできない

```
body {  
  --bg: background-color;  
  var(--bg): pink;  
}
```

プロパティ名を変数にできない



```
body {
```

```
--bg: background-color;
```

```
var(--bg): pink;
```

```
}
```

＼ CSS変数を使う時の注意点 ③ ／

呼び出す時に単位を
書き足しても無効となる

呼び出す時に単位を書き足しても無効となる

```
div {  
  --gutter: 30;  
  margin: var(--gutter)px;  
}
```

呼び出す時に単位を書き足しても無効となる



```
div {
```

```
  --gutter: 30;
```

```
  margin: var(--gutter)px;
```

```
}
```


呼び出す時に単位を書き足しても無効となる



```
div {
```

```
  --gutter: 30px;
```

```
  margin: var(--gutter);
```

```
}
```

呼び出す時に単位を書き足しても無効となる



```
div {
```

```
  --gutter: 30;
```

```
  margin: calc(var(--gutter) * 1px);
```

```
}
```

対応ブラウザ

Chrome	Safari	Firefox	Edge	IE
 49+	 9.1+	 31+	 16+	

Polyfill

- **css-vars-polyfill**

<https://github.com/jhildenbiddle/css-vars-polyfill>

- **CSS Variables Polyfill**

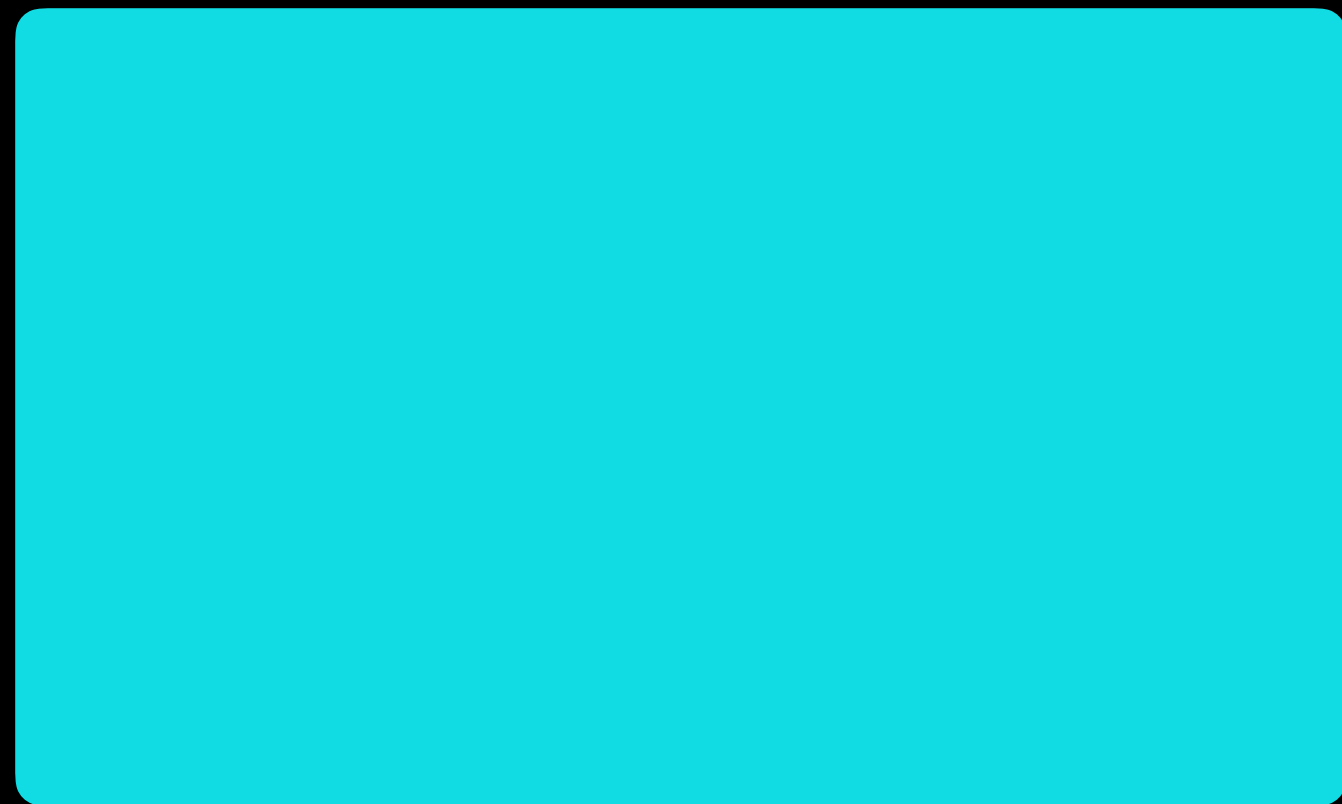
<https://github.com/aaronbarker/css-variables-polyfill>

CSS 小技 テクニ ック 集

＼ マスターしたいCSSテクニック ① ／

Flexbox でレイアウト組み

基本の横並び



基本の横並び

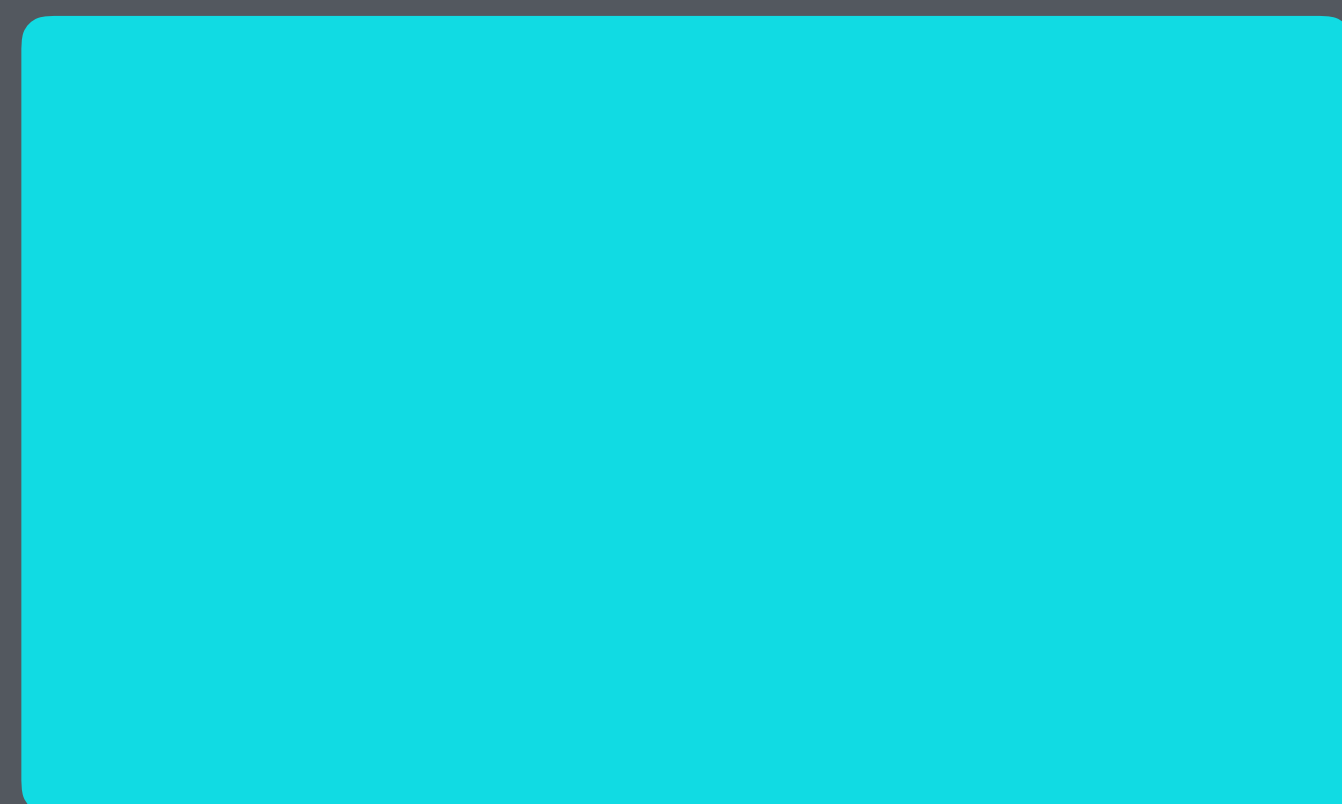
{

- `display: flex;`
- `flex: 1;`

}

Demo

ボックスを上下左右中央に配置

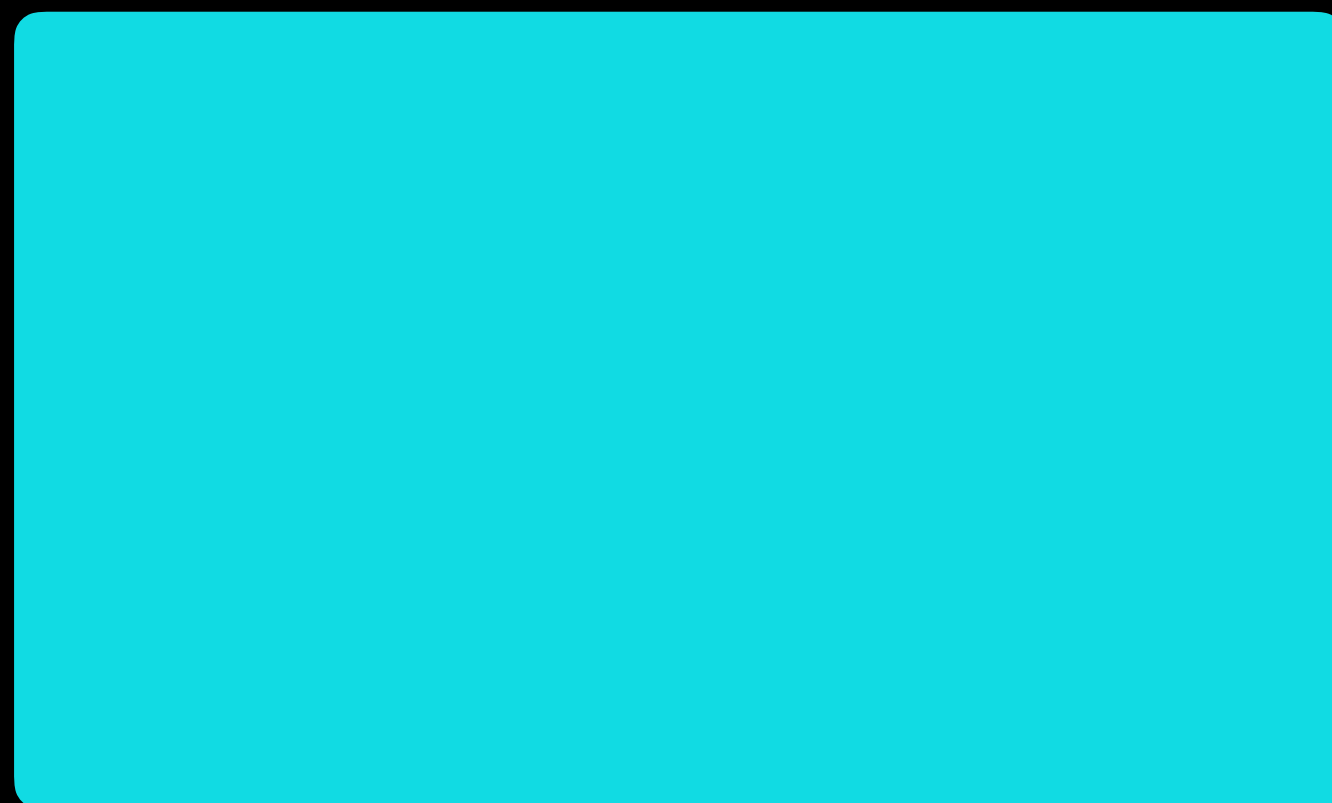


ボックスを上下左右中央に配置

- `display: flex;`
- `justify-content: center;`
- `align-items: center;`

Demo

画像とテキストを互い違いに表示



Five horizontal gray bars of varying lengths, stacked vertically, representing text lines. The bars are positioned to the right of the cyan image block.

Five horizontal gray bars of varying lengths, stacked vertically, representing text lines. The bars are positioned on the left side of the bottom section.








画像とテキストを互い違いに表示

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

Demo

対応ブラウザ

Chrome	Safari	Firefox	Edge	IE
 4+	 3.1+	 2+	 12+	 10+

Polyfill

- `flexibility.js`
<https://github.com/jonathantneal/flexibility>

＼ マスターしたいCSSテクニック ② ／

画像のトリミング

画像のトリミング



画像のトリミング

{
• `object-fit: cover;`
}

Demo

余白を表示させる場合








余白を表示させる場合

{
• object-fit: contain;
}

Demo

対応ブラウザ

Chrome	Safari	Firefox	Edge	IE
 31+	 10+	 36+	 16+	

Polyfill

- `object-fit-images`

<https://github.com/bfred-it/object-fit-images>

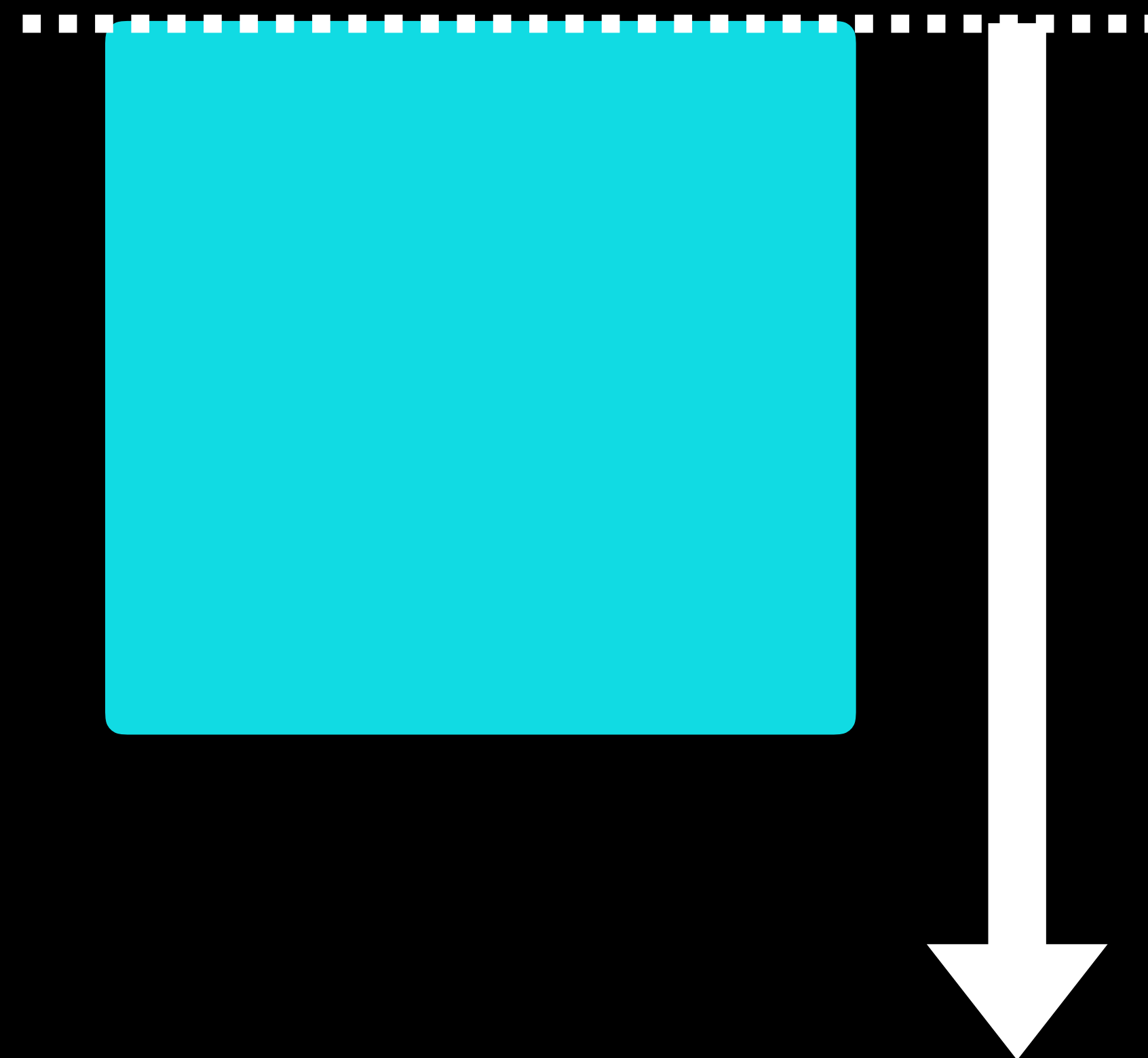
- `object-fit-polyfill`

<https://github.com/constancecchen/object-fit-polyfill>

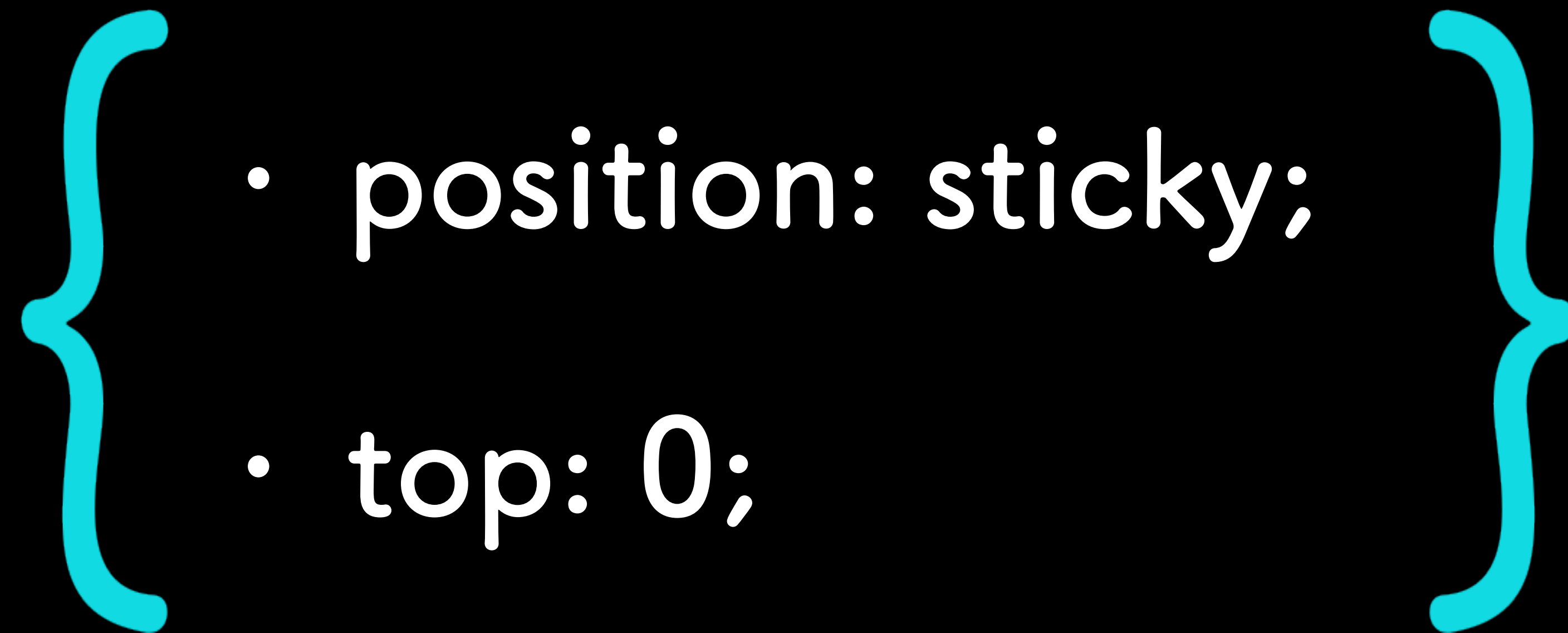
＼ マスターしたいCSSテクニック ③ ／

スクロールすると要素を固定

スクロールすると要素を固定



スクロールすると要素を固定



- `position: sticky;`
- `top: 0;`

Demo

対応ブラウザ

Chrome	Safari	Firefox	Edge	IE
 56+	 7.1+	 32+	 16+	

Polyfill

- **StickyBits**
<https://github.com/dollarshaveclub/stickybits>
- **stickyfill**
<https://github.com/wilddeer/stickyfill>

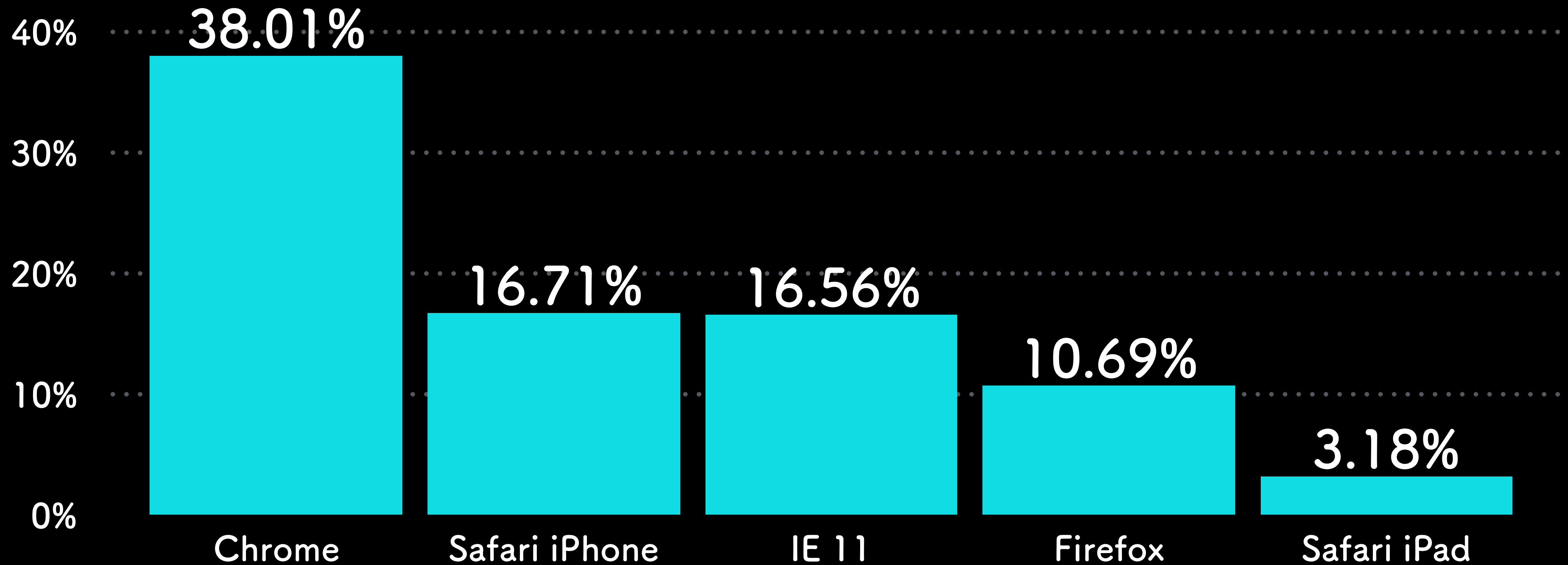
最新テクニック

VS

対応ブラウザ

ブラウザシェア (日本)

gs.statcounter.com



Can I Use...

caniuse.com

Can I use object-fit ? Settings

1 result found

Detected your country as "Japan". Would you like to import usage data for that country?

Import

No thanks

CSS3 object-fit/object-position - CR

Global 88.84% + 1.31% = 90.15%

unprefixed: 86.04% + 1.31% = 87.35%

Method of specifying how an object (image or video) should fit inside its box. object-fit options include "contain" (fit according to aspect ratio), "fill" (stretches object to fill) and "cover" (overflows box but maintains ratio), where object-position allows the object to be repositioned like background-image does.

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			59			10.2		4.4	
	14	54	60	10.1		10.3		4.4.4	
11	15	55	61	11	47	11	all	56	61
	16	56	62	TP	48				
		57	63		49				
		58	64						

Notes

Known issues (0)

Resources (6)

Feedback

Polyfillの是非



- ・ 主要ブラウザに対応できる
- ・ 手間が増える
- ・ 負荷が増える可能性も



TECHACADEMY

techacademy.jp

TECHACADEMY

Webデザイン



WordPress



JavaScript



Webディレクション



Webマーケティング



Ruby



PHP



iOS



Android



Unity



 **TECH** ACADEMY

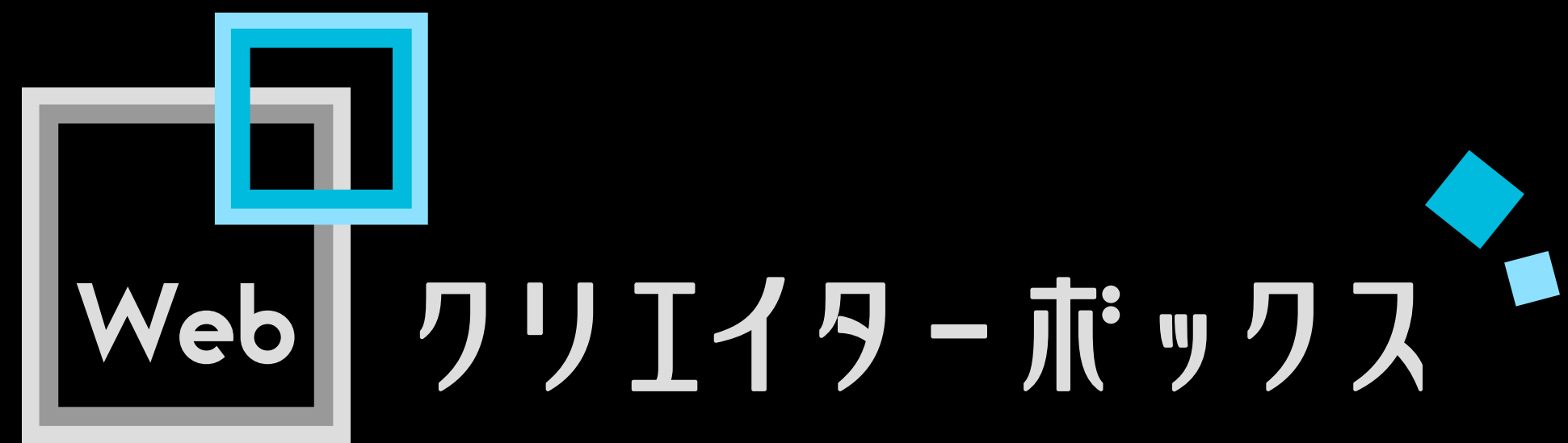
全コース 10,000円引き 

クーポンコード

jun65rg8b

※申込時、備考欄にコードを入力してください

ありがとうございました！



webcreatorbox.com