

@ フロントエンドカンファレンス北海道2025

# オレオレWeb名刺 作っちゃおうぜ

Date : 2025/9/6

Presenter : kCat/@kcat\_fun

Future Univeristy Hakodate 公立はこだて未来大学 FUN



Koichi Kaneko  
金子 康一

Mail : [✉ contact@kcat.dev](mailto:contact@kcat.dev)  
X : [@kcat\\_fun](https://x.com/kcat_fun)  
Skills : [🔧](#) Robotics, Web Frontend



Portfolio



# 00

## 自己紹介



名前 : 金子 康一 / kCat

所属 : 公立はこだて未来大学 4年

趣味 : Robotics, Web Frontend

Web : <https://portfolio.kcat.dev>

✕ : @kcat\_fun

# Agenda | 目次



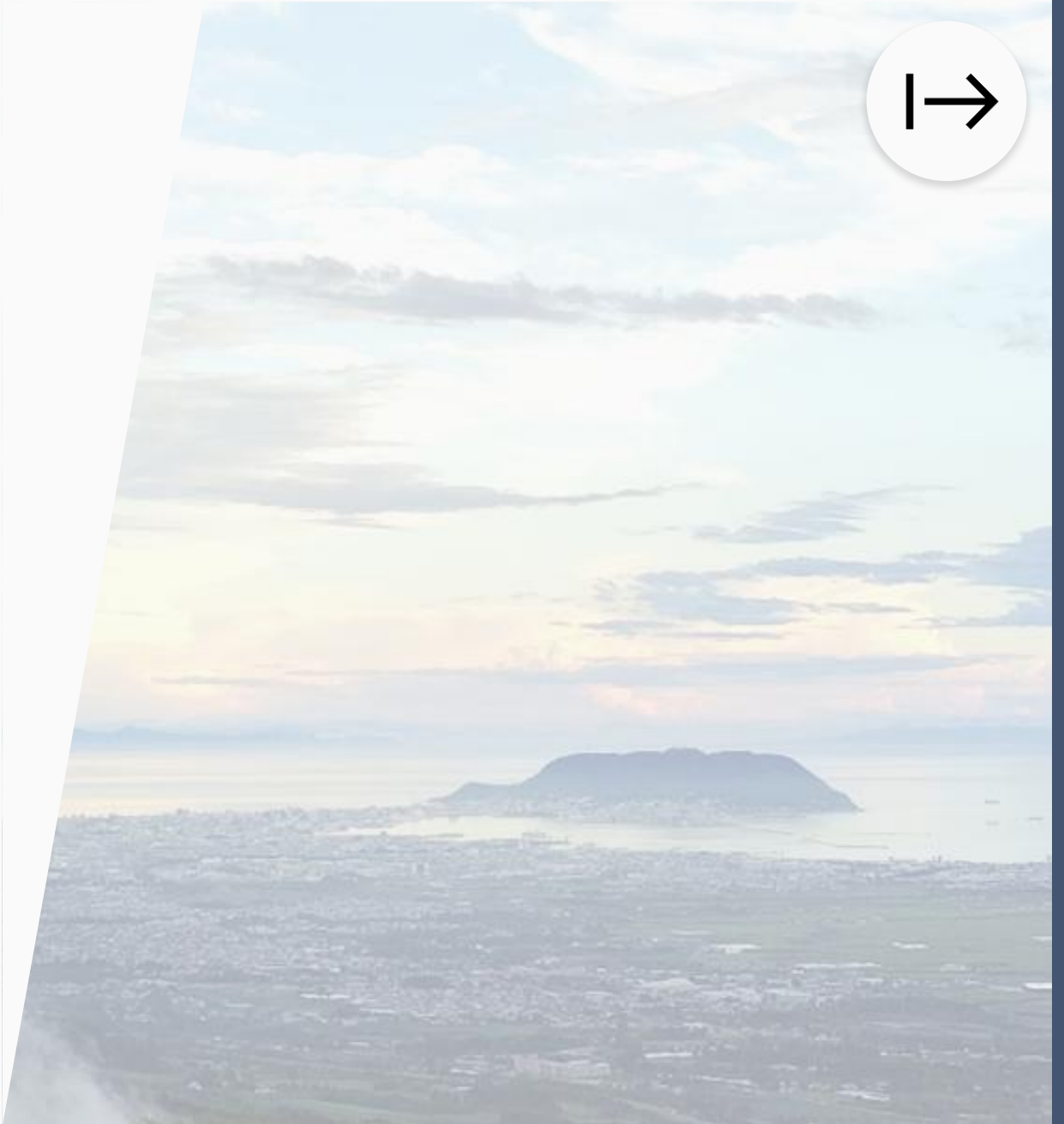
↳ 01 はじめに

<> 02 実装について

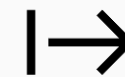
📦 03 立体表示の比較

🚩 04 さいごに

# 01 | はじめに

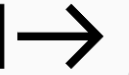


# 01 | はじめに



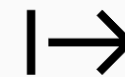
名刺持ってますか？

# 01 | はじめに



- 忘れてきてしまう
- 切らしてしまおう

# 01 | はじめに



- Web名刺で解決！
- せっかくなら  
インタラクティブに！

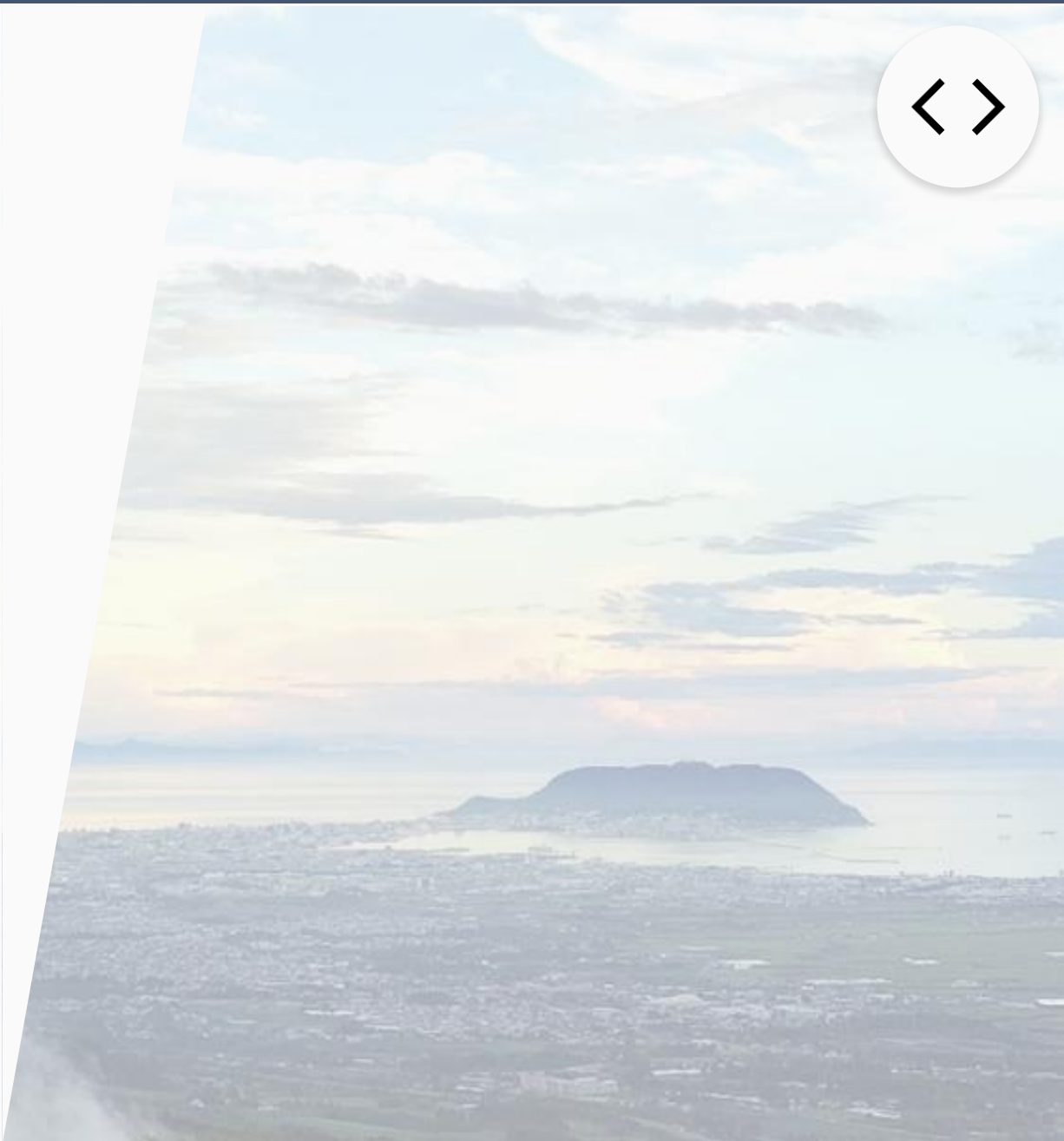
# 01 | はじめに



Web名刺を見る



# 02 | 実装について



# 02

## 実装について



Vanilla環境 is best!!

# 02

## 実装について



## 立体的な回転の実装

# 02

## 実装について



1. クリックした時のマウス座標を記録
2. 1で記録した座標と現在の座標の差で回転
3. 離れたときに自動で正面に向く

# 02

## 実装について



```
// マウスイベント
card.addEventListener("mousedown", startDrag);
document.addEventListener("mousemove", moveDrag);
document.addEventListener("mouseup", endDrag);

// タッチイベント (モバイル対応)
card.addEventListener("touchstart", startDrag);
document.addEventListener("touchmove", moveDrag);
document.addEventListener("touchend", endDrag);
```

# 02

## 実装について



```
let deltaX = currentX - startX;  
let deltaY = (currentY - startY) * Math.max(1.0 - Math.abs(currentTiltX) / 5.0, 0.1);
```

1

// Y軸回転 (左右) は 0~359° の範囲で更新

```
currentRotation = normalizeAngle(currentRotation + deltaX * 0.5);
```

// X軸回転 (上下) は -30° ~ 30° に制限

```
currentTiltX = clampTilt(currentTiltX - deltaY * 0.3);
```

```
frontCard.style.transform
```

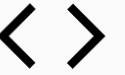
```
  = `rotateY(${currentRotation}deg) rotateX(${currentTiltX}deg)`;
```

```
backCard.style.transform
```

```
  = `rotateY(${180 + currentRotation}deg) rotateX(${ -currentTiltX}deg) translateY(-100%)`;
```

# 02

## 実装について



```
let deltaX = currentX - startX;  
let deltaY = (currentY - startY) * Math.max(1.0 - Math.abs(currentTiltX) / 5.0, 0.1);
```

```
// Y軸回転（左右）は 0~359° の範囲で更新  
currentRotation = normalizeAngle(currentRotation + deltaX * 0.5);  
  
// X軸回転（上下）は -30° ~ 30° に制限  
currentTiltX = clampTilt(currentTiltX - deltaY * 0.3);
```

2

```
frontCard.style.transform  
  = `rotateY(${currentRotation}deg) rotateX(${currentTiltX}deg)`;  
backCard.style.transform  
  = `rotateY(${180 + currentRotation}deg) rotateX(${ -currentTiltX}deg) translateY(-100%)`;
```

# 02

## 実装について



```
let deltaX = currentX - startX;
let deltaY = (currentY - startY) * Math.max(1.0 - Math.abs(currentTiltX) / 5.0, 0.1);

// Y軸回転（左右）は 0~359° の範囲で更新
currentRotation = normalizeAngle(currentRotation + deltaX * 0.5);

// X軸回転（上下）は -30° ~ 30° に制限
currentTiltX = clampTilt(currentTiltX - deltaY * 0.3);
```

```
frontCard.style.transform
  = `rotateY(${currentRotation}deg) rotateX(${currentTiltX}deg)`;
backCard.style.transform
  = `rotateY(${180 + currentRotation}deg) rotateX(${-currentTiltX}deg) translateY(-100%)`;
```

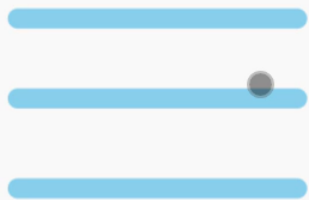
# 02

## 実装について



CSSプロパティ：transform

…動かしたり回転したりできる



例：ハンバーガーメニュー

# 02

## 実装について



```
let deltaX = currentX - startX;
let deltaY = (currentY - startY) * Math.max(1.0 - Math.abs(currentTiltX) / 5.0, 0.1);

// Y軸回転 (左右) は 0~359° の範囲で更新
currentRotation = normalizeAngle(currentRotation + deltaX * 0.5);

// X軸回転 (上下) は -30° ~ 30° に制限
currentTiltX = clampTilt(currentTiltX - deltaY * 0.3);
```

```
frontCard.style.transform
  = `rotateY(${currentRotation}deg) rotateX(${currentTiltX}deg)`;
backCard.style.transform
  = `rotateY(${180 + currentRotation}deg) rotateX(${ -currentTiltX}deg) translateY(-100%)`;
```

3

JavaScriptで動的に回転角度を変更

# 02

実装について



名刺のデザインはどう作るの？

# 02

## 実装について



html, cssで頑張ります

# 02

## 実装について



```
14 <div class="card">
15   <div class="front card-style">
16     <div class="front-card-content">
17       <div class="front-card-header">
18         <div class="college-name-en">Future University Hakodate</div>
19         <div class="college-name-ja">公立はこだて未来大学</div>
20         <div class="fun-icon">FUN</div>
21       </div>
22       <div class="front-tool-icon">
23         
24       </div>
25       <div class="front-name-block">
26         <div class="front-name-en">Kaneko □□□ Koichi</div>
27         <div class="front-name-ja">金子□康一</div>
28       </div>
29       <div class="front-info-block">
30         <ul>
31           <li>
32             <div class="item-name">Mail</div>
33             <div class="colon">:</div>
34             
35             <a href="mailto:contact@kcat.dev" class="item">contact@kcat.dev</a>
36           </li>
37           <li>
38             <div class="item-name">X</div>
39             <div class="colon">:</div>
40             
41             <a href="https://x.com/kcat_fun" class="item">@kcat_fun</a>
42           </li>
```

```
79 .front {
80   position: relative;
81   transform-origin: center;
82   background-color: #FAFAFA;
83   transform: rotateY(0deg);
84
85   .front-right-block {
86     position: absolute;
87     top: 0;
88     right: 0;
89     display: block;
90     width: 2.5%;
91     height: 100%;
92     background-color: #010555;
93   }
94
95   .front-card-content {
96     font-size: 100%;
97     position: relative;
98     width: 97%;
99     height: 100%;
100    margin-right: 2.5%;
101
102    .front-card-header {
103      position: relative;
104      font-size: 100%;
105      display: flex;
106      justify-content: right;
107      align-items: end;
108      margin: 0 0 0 auto;
109      padding-top: 3%;
110      width: 90%;
```

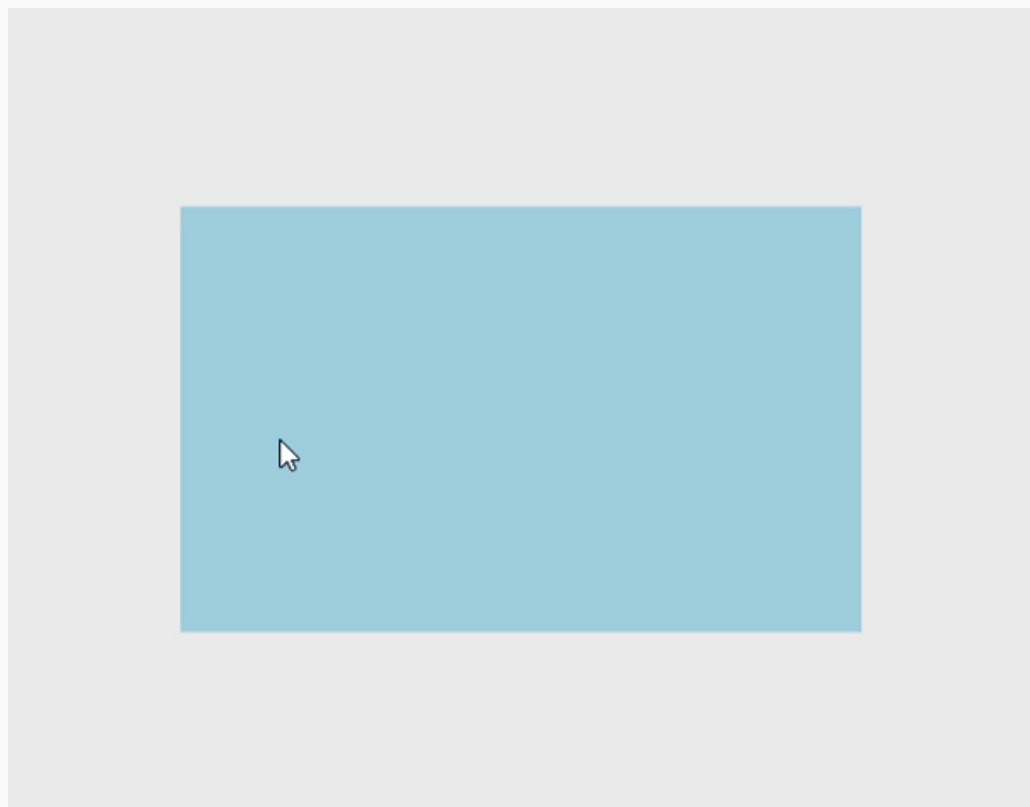
03

立体表示の比較



# 03

## 立体表示の比較



**p5.js version**

光の反射加減がリアル

# 03

## 立体表示の比較



### Three.js version

光の反射加減がよりリアル  
影がリアルに投影

# 03

## 立体表示の比較



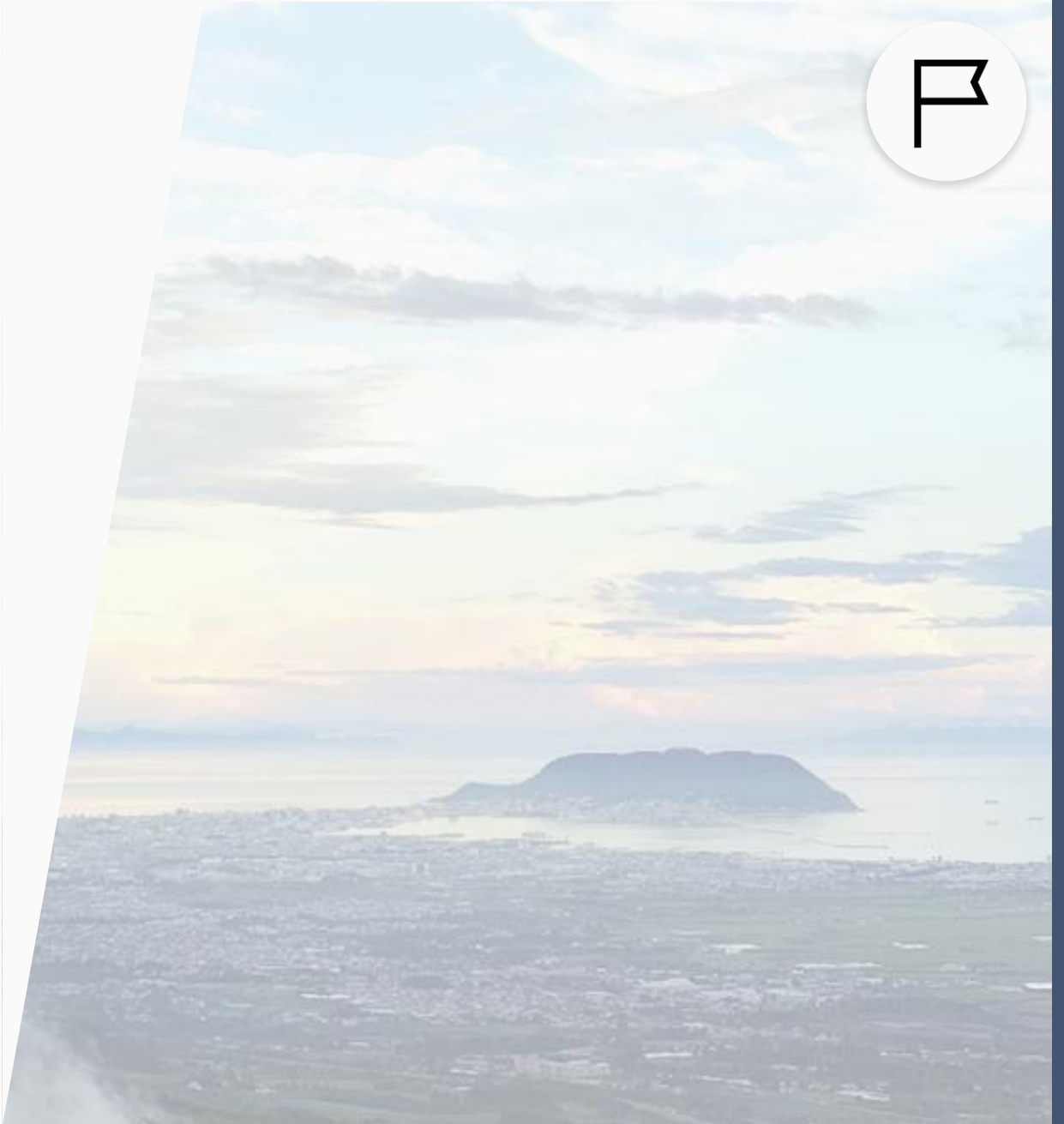
### メリット

- リアルな表示ができる

### デメリット

- htmlやcssで文字や画像を表示したり調整できない
  - jsで追加し、座標を調整する必要がある
    - レスポンシブにしたり細かな調整が面倒

# 04 | さいごに



# 04

さいごに



## 謝罪

完全Vanillaといいましたが嘘です

モジュールバンドラにParcelを使用しています

# 04

## さいごに



# 04 | さいごに



Web名刺

<https://github.com/kCat-fun/web-meishi>

ライブラリ比較

<https://github.com/kCat-fun/web-meishi-library-comparison>

# 04 | さいごに



ご清聴ありがとうございました