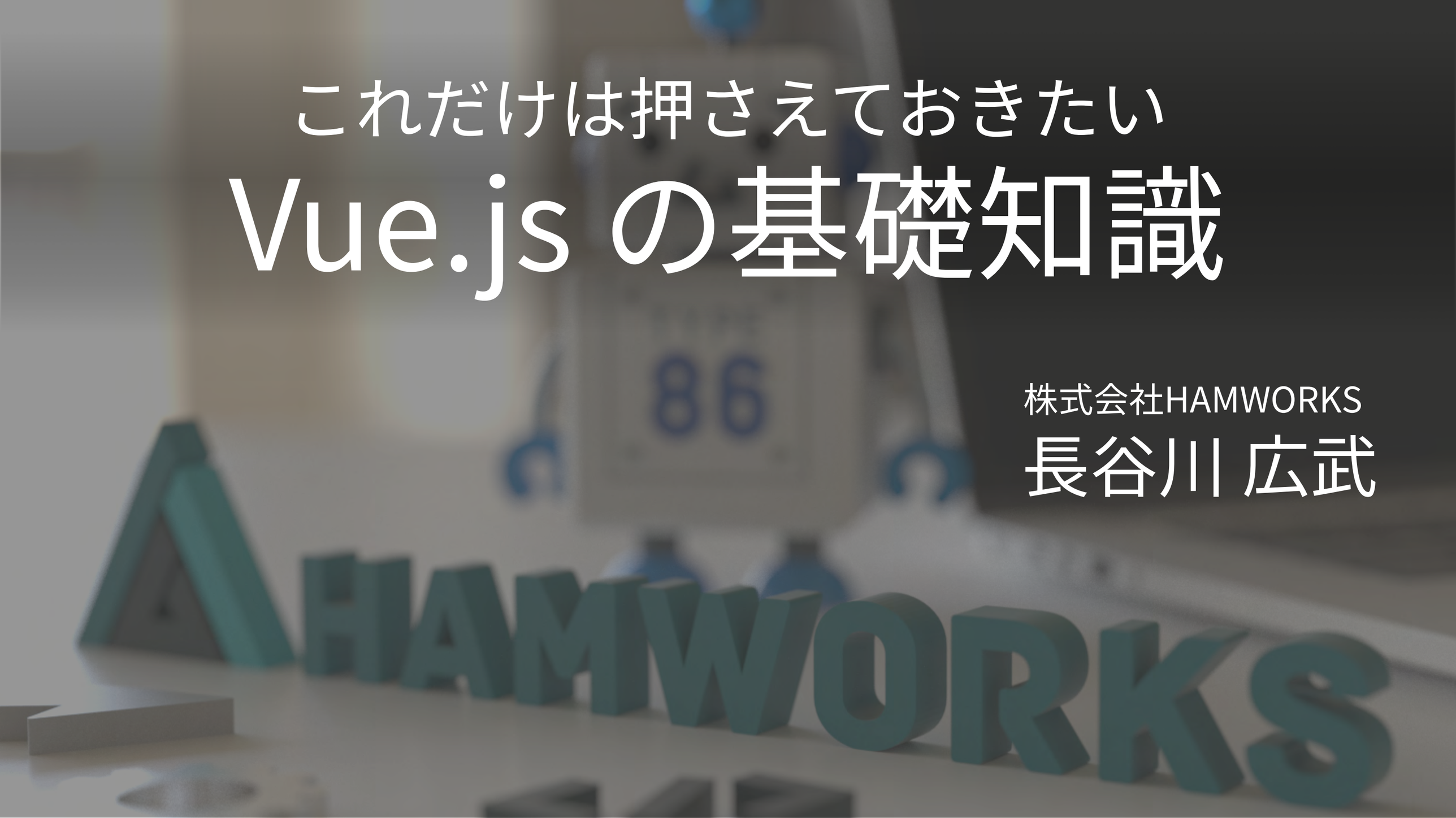
CSS Nite LP64

Coder's High 2019

2019.10.19



これだけは押さえておきたい

Vue.js の基礎知識

株式会社HAMWORKS
長谷川 広武

自己紹介


はせがわ　ひろむ
長谷川 広武

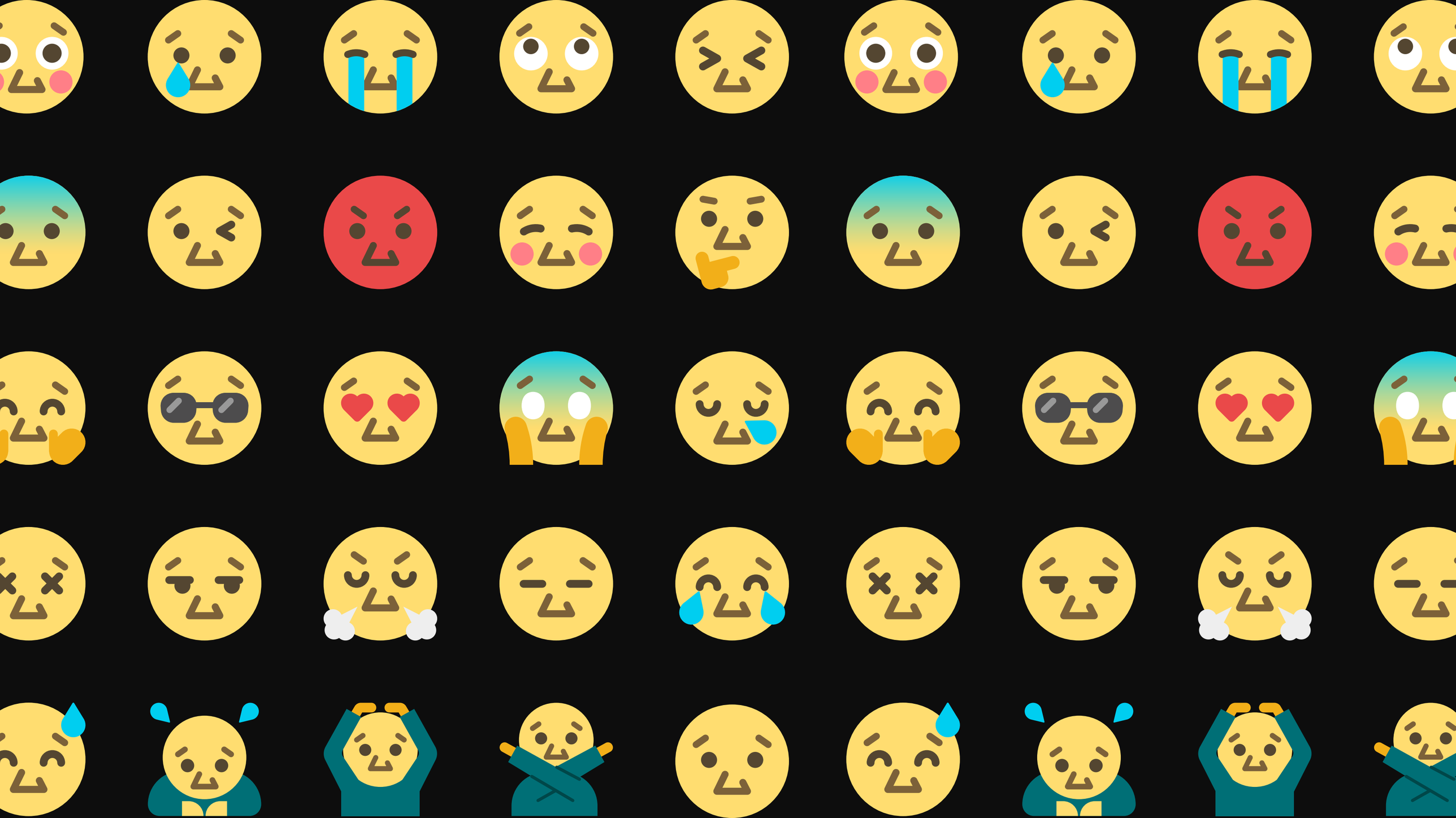
ハム

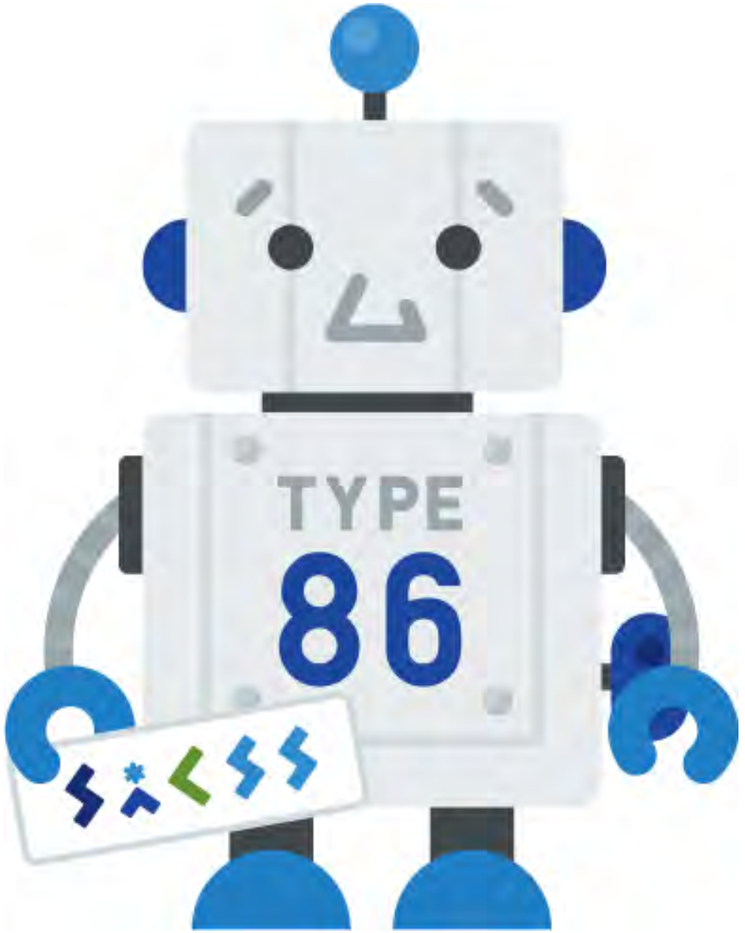
株式会社HAMWORKS

bit part 合同会社

フロントエンドエンジニア
テクニカルディレクター

SaCSS 主催




TYPE
86

ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW
°`) HAMWORKS (´°ム°`) HAMWORKS (´°
ORKS (´°ム°`) HAMWORKS (´°ム°`) HAMW

HAMWORKS

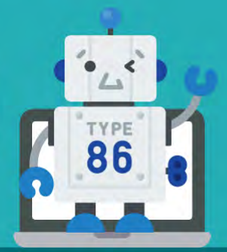
デザインからコーディング、各種CMSの実装まで

制作に関わる仕事をサポート!

DESIGN
デザイン

CMS
各種CMS実装

CODING
コーディング



「こんなのできる?」
などのご相談も
お待ちしております (´°ム°`)

株式会社HAMWORKS
https://ham.works
info@ham.works
hamworks.co.ltd


Sapporo.css


ウェブマーケター
多言語リスティング+ウェブサイト制作・アクセス解析)
プロジェクトマネージャー
ブランディング・マーケティング・多言語予約システム・食品ECなど)
語学留学&ウェブデザイナー・エンジニア
n Franciscoにて)
Tools
Evernote
OmniFocus
Wunderlist
Trello

本題

Vue.js

Learn Ecosystem Team Support Vue Translations

The Progressive JavaScript Framework

WHY VUE.JS? GET STARTED GITHUB

Special Sponsor

stdlib

Function as a Service Platform and Library

Approachable

Already know HTML, CSS and JavaScript? Read the guide and start building things in no time!

Versatile

An incrementally adoptable ecosystem that scales between a library and a full-featured framework.

Performant

20KB min+gzip Runtime
Blazing Fast Virtual DOM
Minimal Optimization Efforts

https://vuejs.org/

似ているだけですよ

The Progressive JavaScript Framework

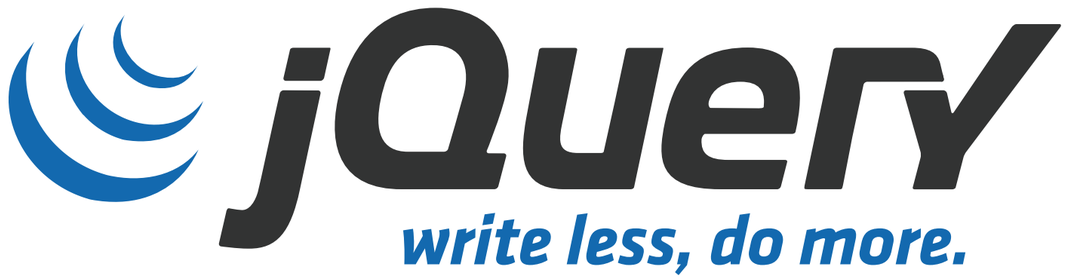
プログレッシブフレームワーク

直訳すれば進歩的なフレームワークですが、

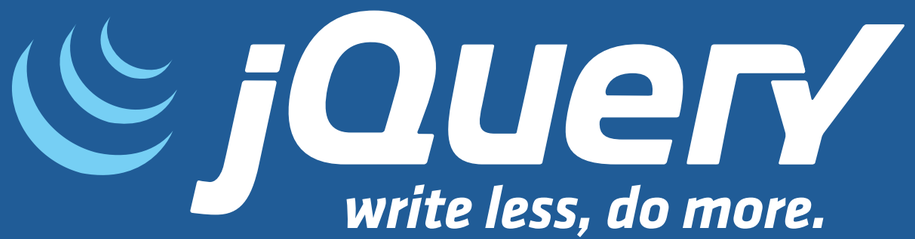
段階的に開発者が機能を選択できる

フレームワークという意味

ちなみに


jQuery
write less, do more.

ライブラリ



ライブラリ

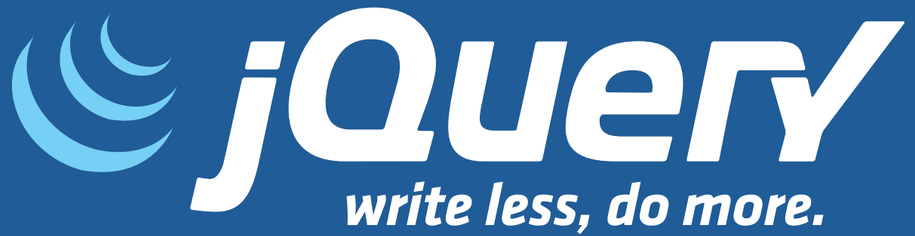
→ 汎用的な機能を
まとめたもの

例：家電

フレームワーク

→ 処理を実行する
ための枠組み

例：家



ライブラリ

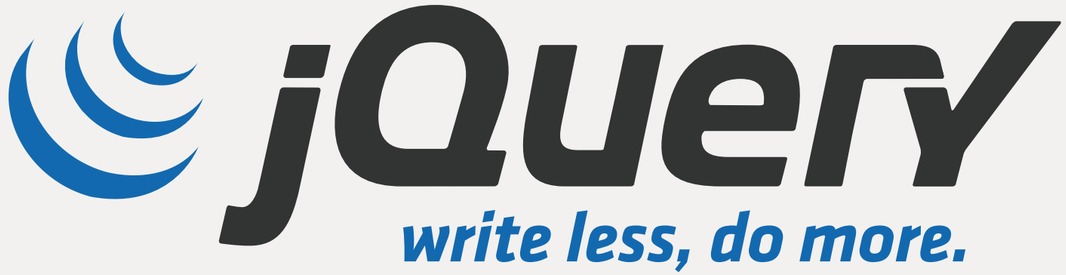
→ 汎用的な機能を
まとめたもの

例：家電

フレームワーク

→ 処理を実行する
ための枠組み

例：家



- ウェブサイト制作の現場では現役
- シンプルなコードで利用可能
- プラグイン利用による効率化は絶大
- DOMの追加にはテンプレート利用

サンプルデータ

```
const items = [
    { title: 'list title01', link: 'entry01.html' },
    { title: 'list title02', link: 'entry02.html' },
    { title: 'list title03', link: 'entry03.html' },
    { title: 'list title04', link: 'entry04.html' }
  ];
```

JavaScriptテンプレート未使用 HTML

```
<button class=“js-show-list">リスト表示</button>
<ul id=“list">
</ul>
```

https://codepen.io/h2ham/pen/BaBowOV?editors=1010

JavaScriptテンプレート未使用 JavaScript

```
<script>
$('.js-show-list').on('click', function() {
  $(‘#list').empty();
  $.each(items, function() {
    $('#list').append('<li>'
      + '<a href="'+ this.link + '">'
      + this.title
      + '</a>'
      + '</li>');
  });
});
</script>
```

https://codepen.io/h2ham/pen/BaBowOV?editors=1010

JavaScriptテンプレートを使うなら

- handlebars.js など

handlebars

Minimal templating on steriods

Get Started →

Semantic templates

Handlebars provides the power necessary to let you build semantic templates effectively with no frustration.

Mustache-compatible

Handlebars is largely compatible with Mustache templates. In most cases it is possible to swap out Mustache with Handlebars and continue using your current templates.

Fast execution

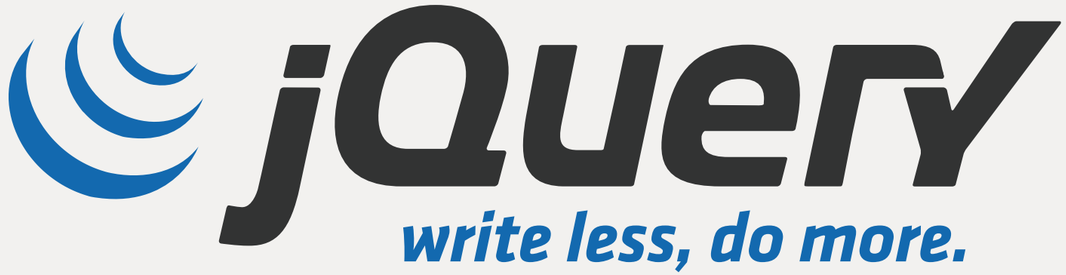
Handlebars compiles templates into JavaScript functions. This makes the template execution faster than most other template engines.

handlebars.js 利用

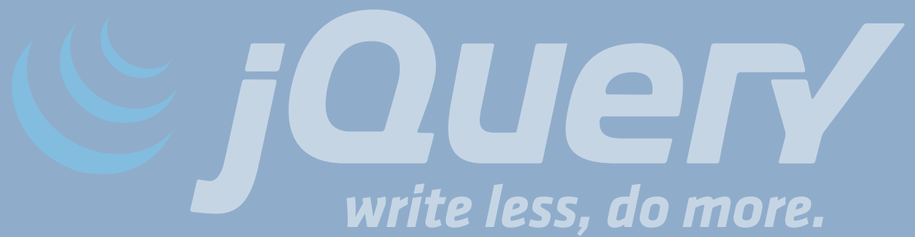
```
<script id="linklist" type="text/x-handlebars-template">
  {{#each this}}
  <li><a href="{{ link }}">{{ title }}</a></li>
  {{/each}}
</script>
<script>
$('.sample-selector').on('click', function() {
  const source = $("#linklist").html();
  const template = Handlebars.compile(source);
  const html = template(items);
  $("#list").html(html);
});
</script>
```

http://codepen.io/h2ham/pen/yarORd

苦手・不向きな制作も
多くなってきた



- ✖ 複雑な機能作成には不向き
- ✖ ウェブアプリでの利用は要検討
- ✖ HTMLにつけている
  JavaScriptのコードが探しづらい
  （推測しづらい）



ライブラリ

→ 汎用的な機能を
まとめたもの

例：家電

フレームワーク

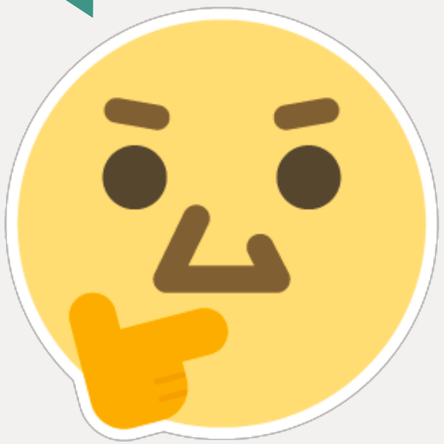
→ 処理を実行する
ための枠組み

例：家

- データと表示を同期させる処理が得意
- DOMの変更周りが楽にできる
- 見通しが良くなる
- 他のフレームワークよりも
  導入の敷居が低い

Vue.js 未体験の人のために

Vue.js に触れてみましょう！



Vue.js の基礎

ファイルの読み込み

ガイド 2.x

基本的な使い方

インストール

Vue Devtools

<script> 直接組み込み

CDN

NPM

CLI

さまざまなビルドについて

用語

ランタイム + コンパイラとランタイム限定の違い

開発モードと本番モードの違い

CSP 環境

開発版のビルド

Bower

AMD モジュールローダ

はじめに

Vue インスタンス

<script> 直接組み込み

ダウンロードし script タグで読み込んでください。 `Vue` はグローバル変数として登録されます。

> ! 開発中は本番バージョンを使用しないでください。 警告や一般的な間違いを見逃す可能性があります!

開発バージョン 警告出力とデバッグモードあり

本番バージョン 警告出力なし、 33.30 KB min+gzip

CDN

プロトタイピングや学習を目的とする場合は、以下のようにして最新バージョンを使うことができます:

```HTML
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

本番環境では、新しいバージョンによる意図しない不具合を避けるため、特定のバージョン番号

HTML

```
<script src="vue.js"></script>
```

HTML

```
<script src="vue.js"></script>
<script>
new Vue({
  el: '#id',
  ~
})
</script>
```

```
<script src="vue.js"></script>
<script>
new Vue({
  el: '#id',
})
</script>
```

el は element の略

Vue.js が作用を及ぼす範囲

HTML

```
<div id="app">
  <p>{{ message }}</p>
</div>

<script src="vue.js"></script>
<script>
new Vue({
  el: '#app',
  ~
})
</script>
```

Hello World の表示

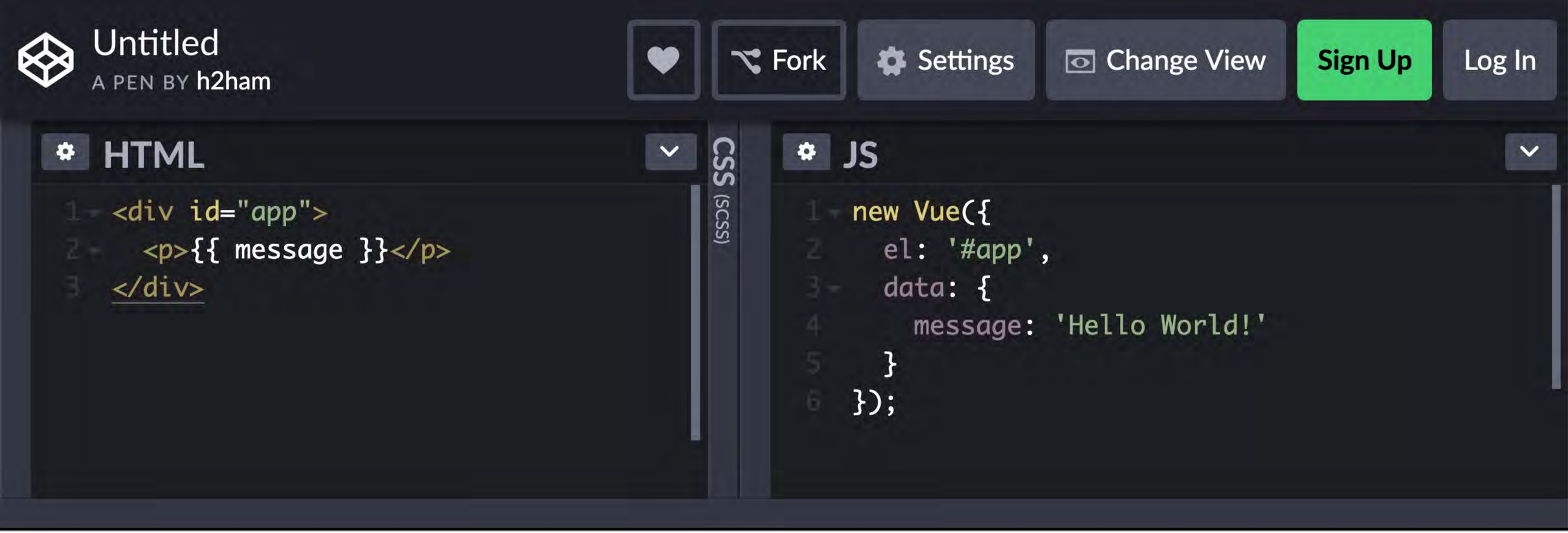
Hello World 文字列の表示

HTML

```
<div id="app">
  <p>{{ message }}</p>
</div>
```

JavaScript

```
new Vue({
  el: '#app',
  data: {
    message: 'Hello World!'
  }
})
```


Untitled
A PEN BY h2ham
Fork
Settings
Change View
Sign Up
Log In
HTML
CSS (SCSS)
JS

{{ message }}

new Vue({
el: '#app',
data: {
message: 'Hello World!'
}
});

Hello World!

Hello World 表示 => テキストの展開

HTML

```
<div id="app">
  <p>{{ message }}</p>
</div>
```

HTML

```
<div id="app">
  <p>{{ message }}</p>
</div>
```

Mustache 構文

(二重中括弧)

Mustache 構文

(二重中括弧)

リアクティブ

HTML

```
<div id="app">
  <p>{{ message }}</p>
</div>
```

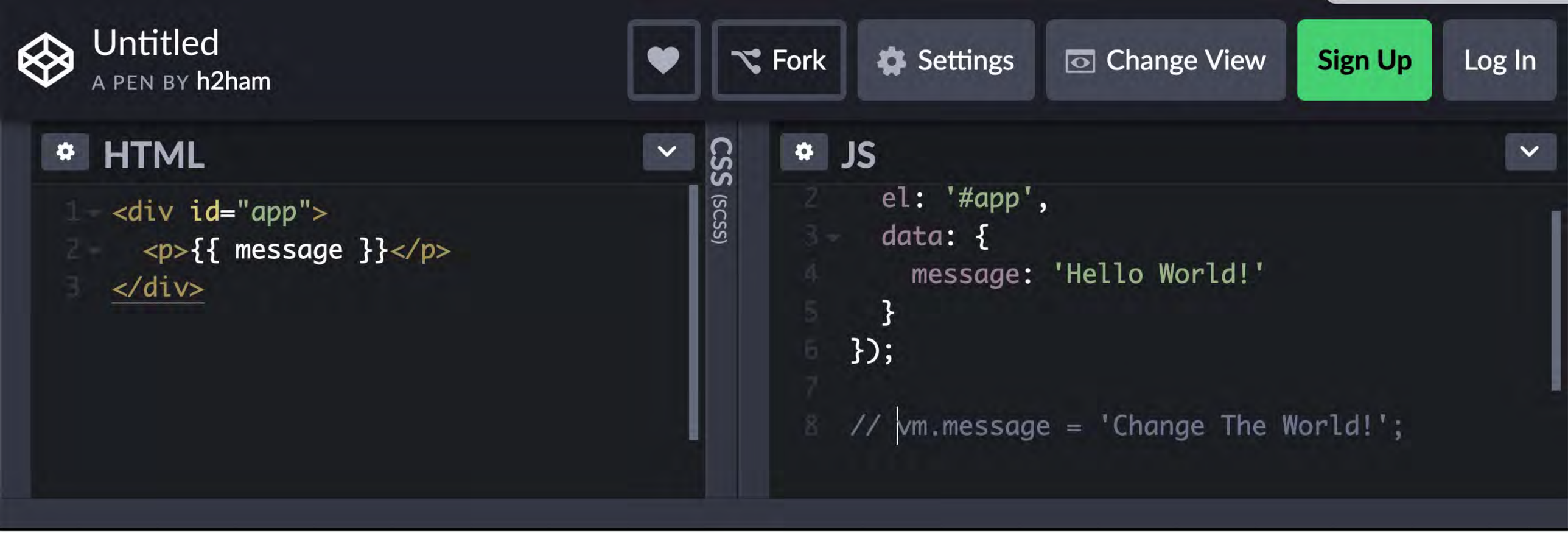
```
message: 'Hello World!'
```

```
<div id="app">
  <p>{{ message }}</p>
</div>
```

```
message: 'Change (´ºДºˋ)'
```

```
const vm = new Vue({
  el: '#app',
  data: {
    message: 'Hello World!'
  }
})

vm.message = 'Change The World!';
```

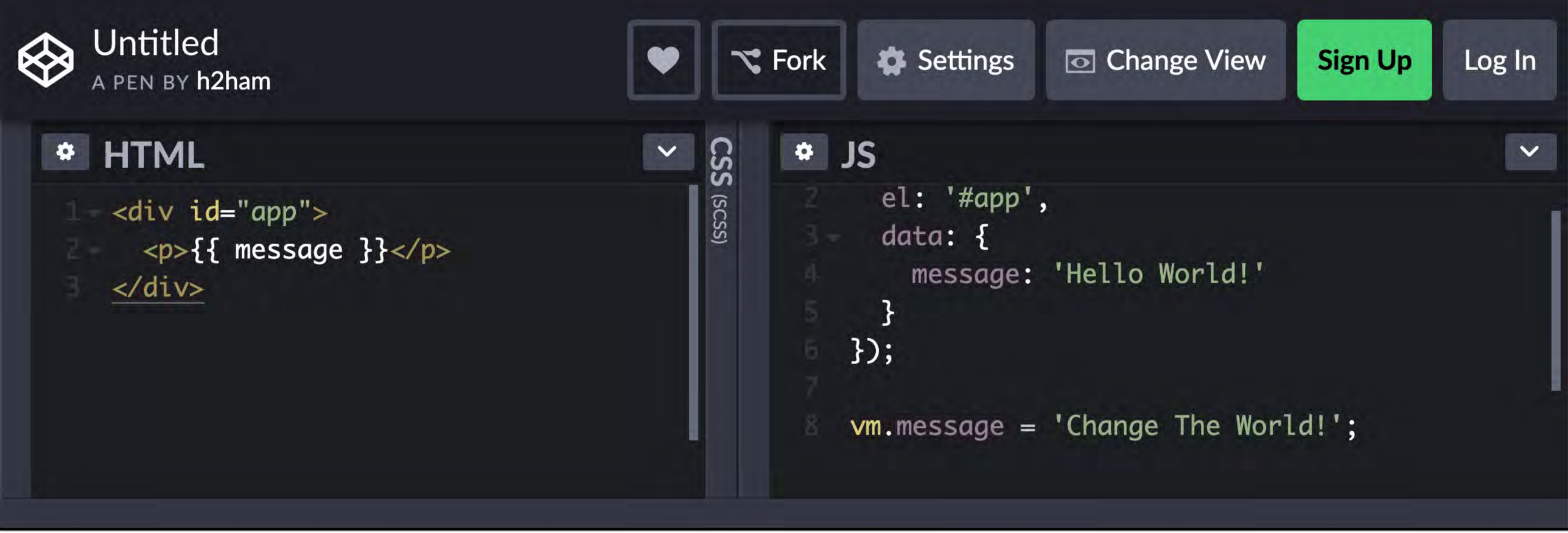


Hello World!

```
    data: {
      message: 'Hello World!'
    }
  });

  // vm.message = 'Change The World!';
```

https://codepen.io/h2ham/pen/yLBYzwm?editors=1010



Change The World!

リアクティブなdataを変更すると表示（View）も変わる

ディレクティブ（v-xxx）

HTML側に書く
Vue.js の構成要素（v-xxx）

条件分岐での表示

```
<div id="app">
  <h1 v-if="isOk">条件分岐v-if OK</h1>
  <h1 v-if="isNo">条件分岐v-if NO</h1>
</div>
```

```
const vm = new Vue({
  el: '#app',
  data: {
    isOk: true,
    isNo: false
  }
})
```

条件分岐v-if OK

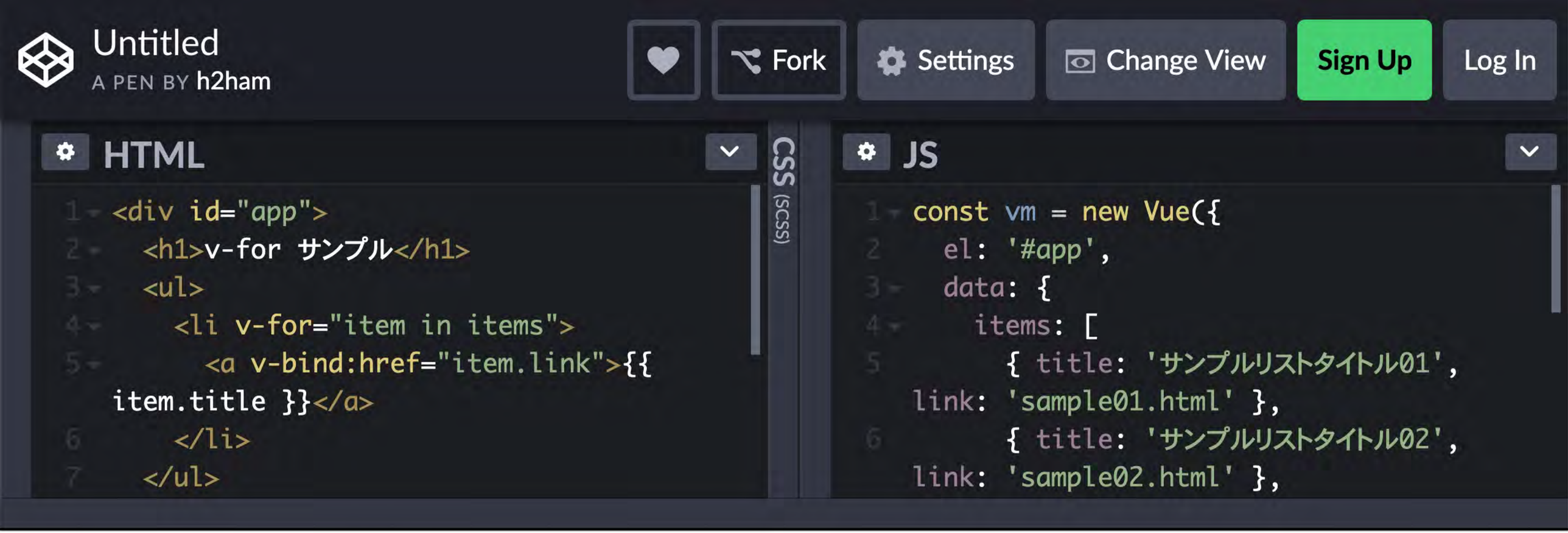
ループ＆属性

HTML

```
<div id="app">
  <h1>v-for サンプル</h1>
  <ul>
    <li v-for="item in items">
      <a v-bind:href="item.link">
        {{ item.title }}
      </a>
    </li>
  </ul>
</div>
```

JavaScript

```
const vm = new Vue({
  el: '#app',
  data: {
    items: [
 { title: 'サンプルリストタイトル01', link: 'sample01.html' },
 { title: 'サンプルリストタイトル02', link: 'sample02.html' },
 { title: 'サンプルリストタイトル03', link: 'sample03.html' },
 { title: 'サンプルリストタイトル04', link: 'sample04.html' }
    ]
  }
})
```



v-for サンプル

- サンプルリストタイトル01
- サンプルリストタイトル02
- サンプルリストタイトル03
- サンプルリストタイトル04

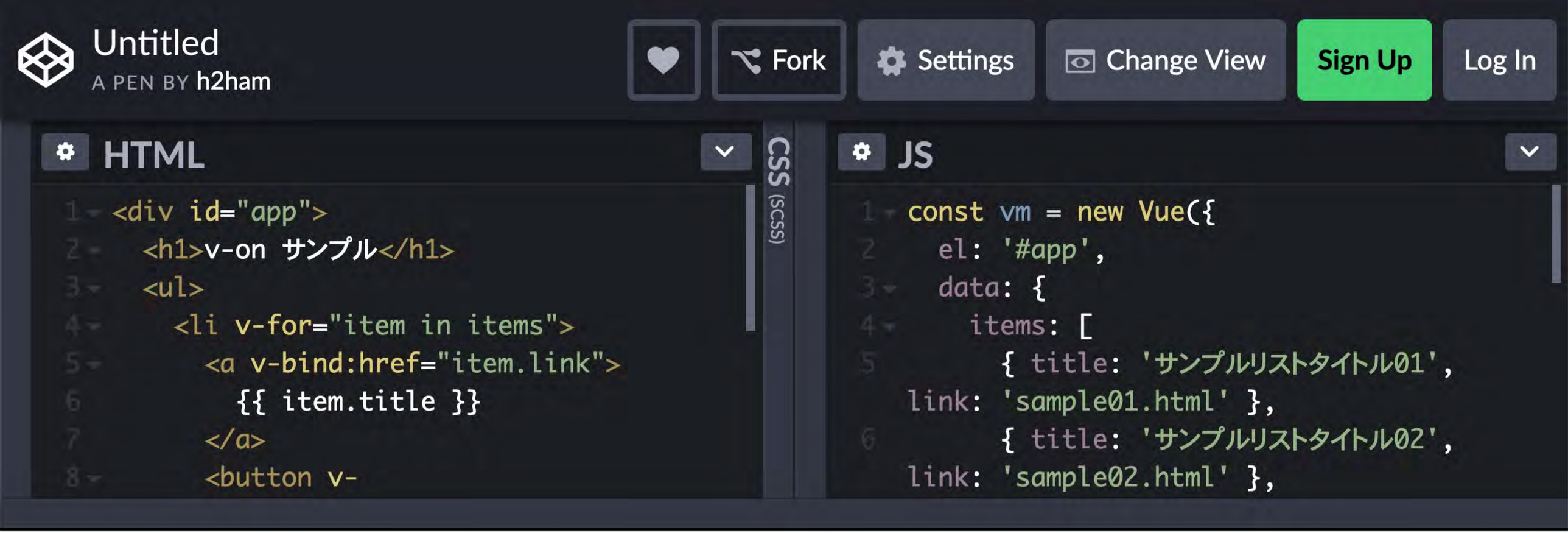
イベント

HTML

```
<div id="app">
  <h1>v-on サンプル</h1>
  <ul>
    <li v-for="item in items">
      <a v-bind:href="item.link">
        {{ item.title }}
      </a>
      <button v-on:click="removeItem(item)">×</button>
    </li>
  </ul>
</div>
```

JavaScript

```
const vm = new Vue({
  el: '#app',
  data: {
    items: [
 { title: 'サンプルリストタイトル01', link: 'sample01.html' },
  ~省略~
    ]
  },
  methods: {
    removeItem: function(item){
      this.items.splice(this.items.indexOf(item), 1);
    }
  }
})
```



v-on サンプル

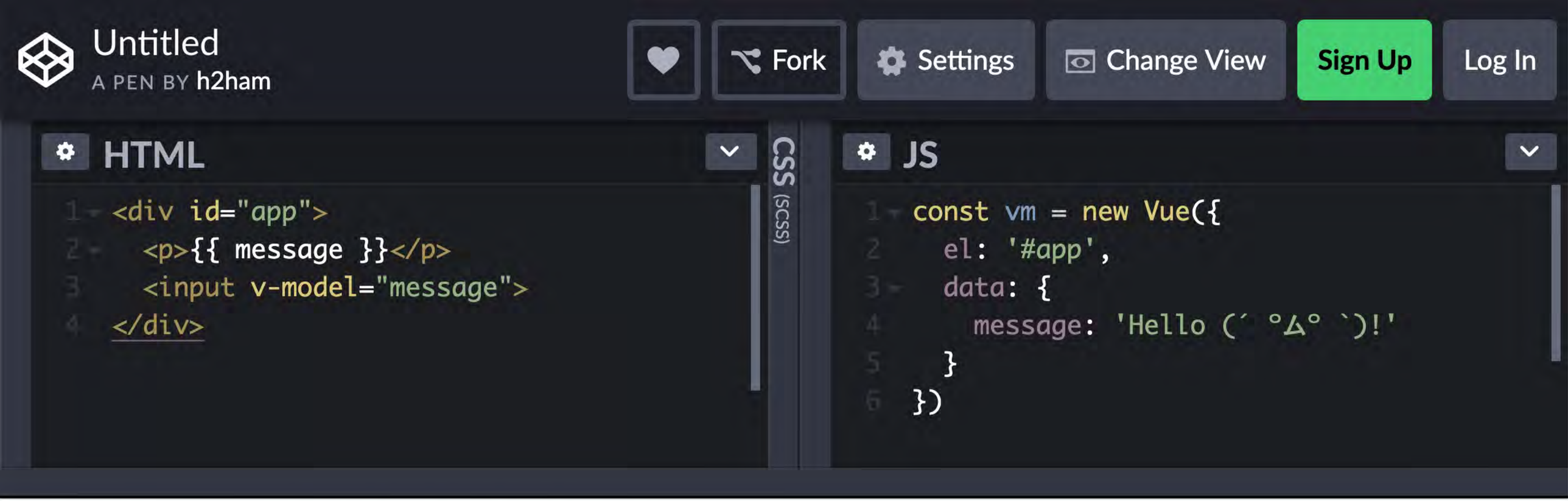
- サンプルリストタイトル01 ×
- サンプルリストタイトル02 ×
- サンプルリストタイトル03 ×
- サンプルリストタイトル04 ×

双方向バインディング

HTML

```
<div id="app">
  <p>{{ message }}</p>
  <input v-model="message">
</div>
```

```
const vm = new Vue({
  el: '#app',
  data: {
    message: 'Hello (´ºДºˋ)!'
  }
});
```



Hello (´ °Д° `)!

Hello (´ °Д° `)!

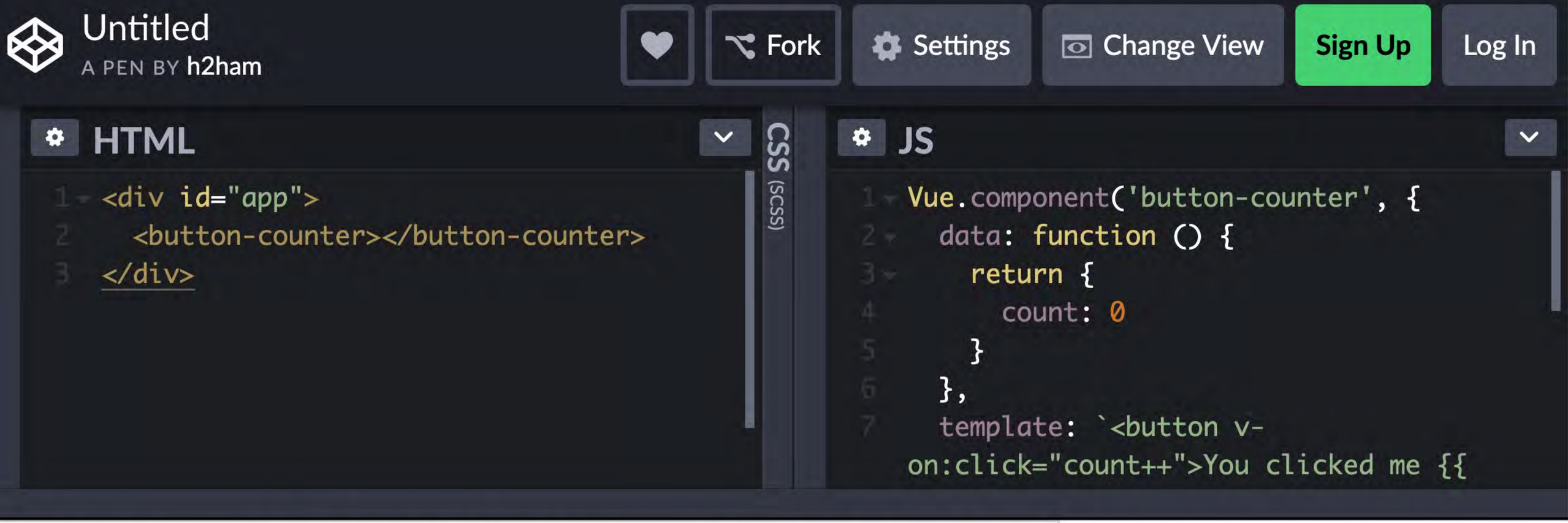
https://codepen.io/h2ham/pen/mdbeZjQ?editors=1010

コンポーネント

HTML

```
<div id="app">
  <button-counter></button-counter>
</div>
```

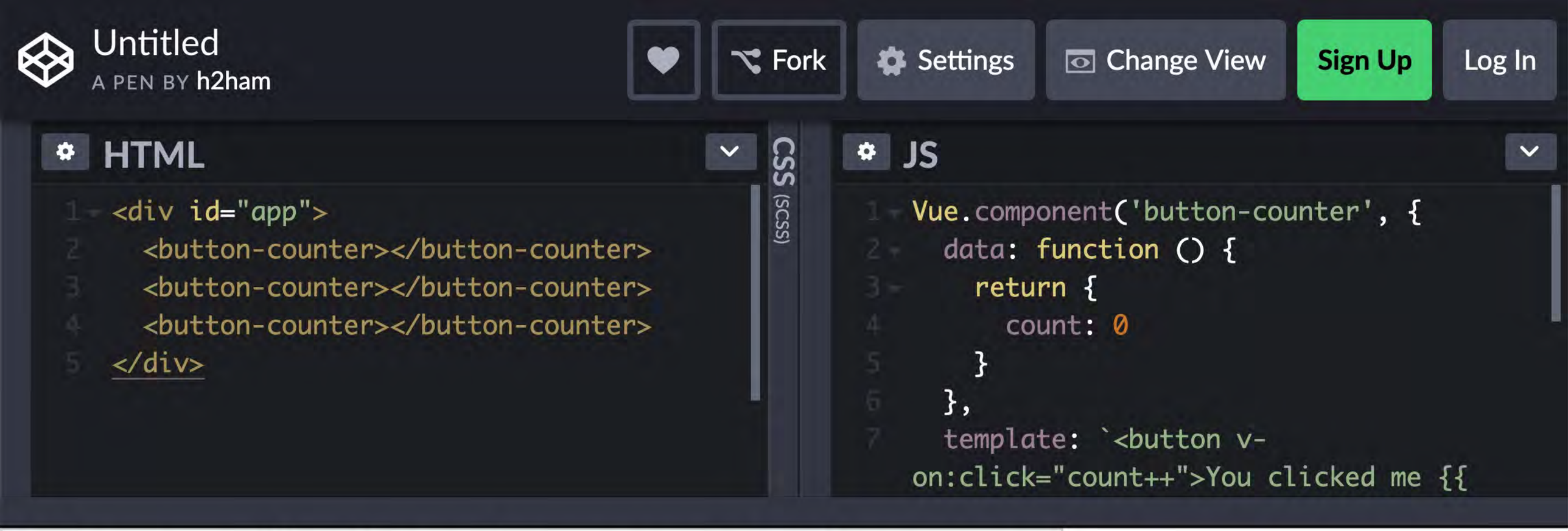
```
Vue.component('button-counter',{
  data: function(){
    return { count: 0 }
  },
  template:
    `<button v-on:click="count++">
      {{ count }}
     </button>`
});
const vm = new Vue({
  el: '#app'
});
```



https://codepen.io/h2ham/pen/eYOJgzN?editors=1010

HTML

```
<div id="app">
  <button-counter></button-counter>
  <button-counter></button-counter>
  <button-counter></button-counter>
</div>
```



You clicked me 0 times.

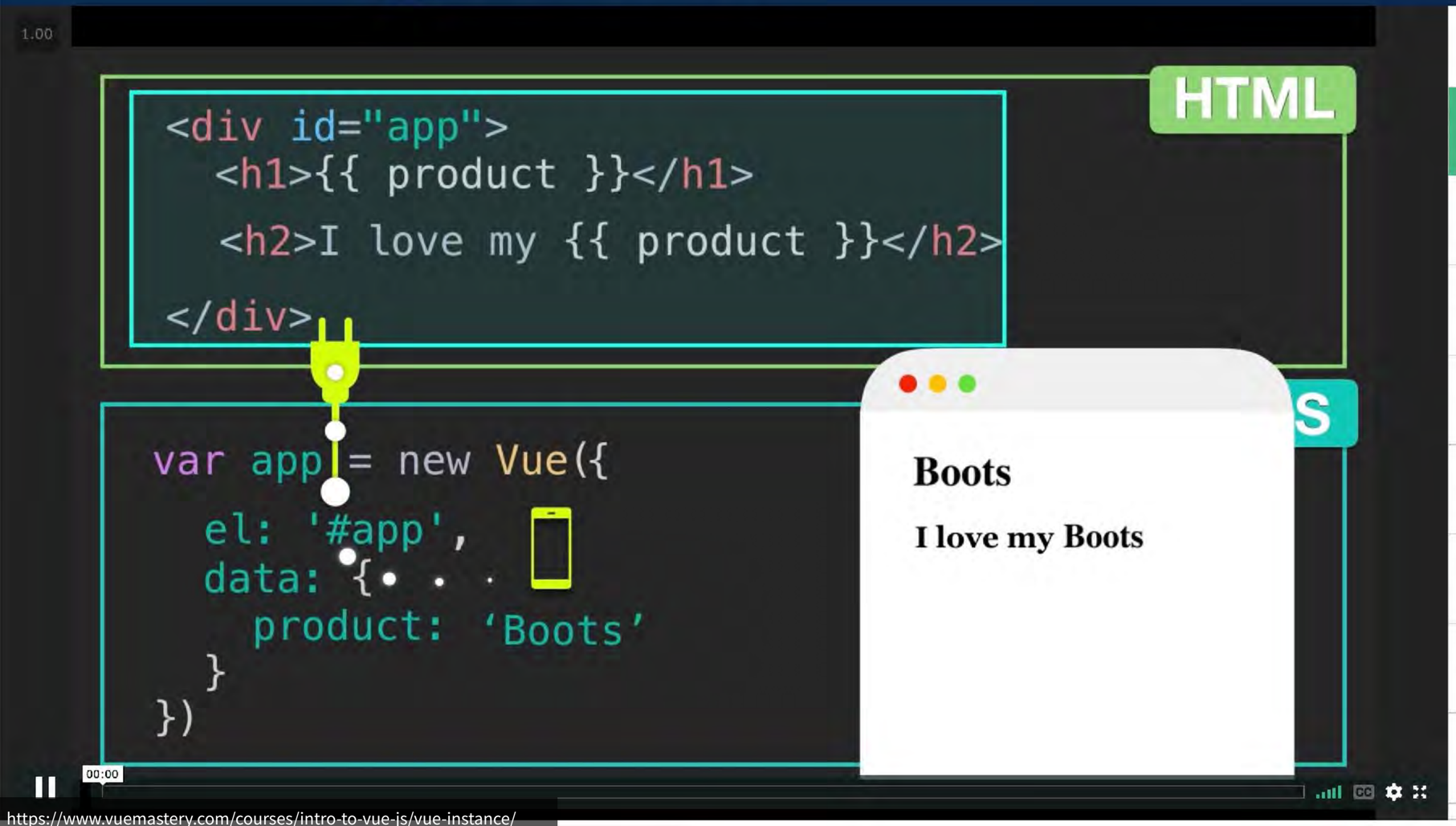
You clicked me 0 times.

You clicked me 0 times.

https://codepen.io/h2ham/pen/bGbEgBO?editors=1010

チュートリアル

Intro to Vue.js



Lessons

1. The Vue Instance
   5:42
2. Attribute Binding
   2:46
3. Conditional Rendering
   3:44
4. List Rendering
   2:16
5. Event Handling
   4:13
6. Class & Style Binding
   5:06
7. Computed Properties
   4:57
8. Components
   6:20

https://www.vuemastery.com/courses/intro-to-vue-js/vue-instance/

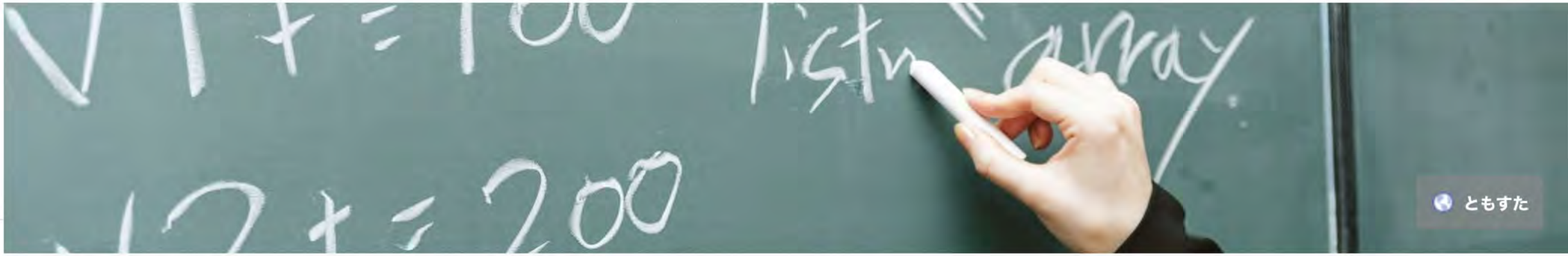
ホーム
急上昇
登録チャンネル
ライブラリ
履歴

動画の評価、コメント、チャンネル登録を行うにはログインしてください。

ログイン

BEST OF YOUTUBE

音楽
スポーツ
ゲーム
映画
テレビ番組



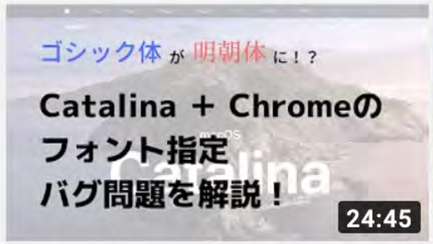
たにぐち まことのプログラミング学習応援チャンネル

チャンネル登録者数 5680人

チャンネル登録

ホーム　動画　再生リスト　コミュニティ　チャンネル　概要

アップロード済み　すべて再生　並べ替え



Catalina Chromeでゴシック体が明朝体に？　フォント...

612 回視聴・3 日前

AWS EC2入門 #01： AWSを始めよう

881 回視聴・4 日前

非SSL混在コンテンツをChromeがブロック。どんな...

593 回視聴・1 週間前

Surface Pro Xを買うのはちょっと待って！？　ARMプロ...

7276 回視聴・1 週間前

Vue.js入門 #01： 一番最初のプログラム

Vue.js入門 #02： 最初のプログラム解説

Vue.js入門 #03：**if**構文とディレクティブ

Vue.js入門 #04：**Vue.js**による属性の書き換え

Vue.js入門 #05：ボタンクリックで現在時刻を表示する **-...**

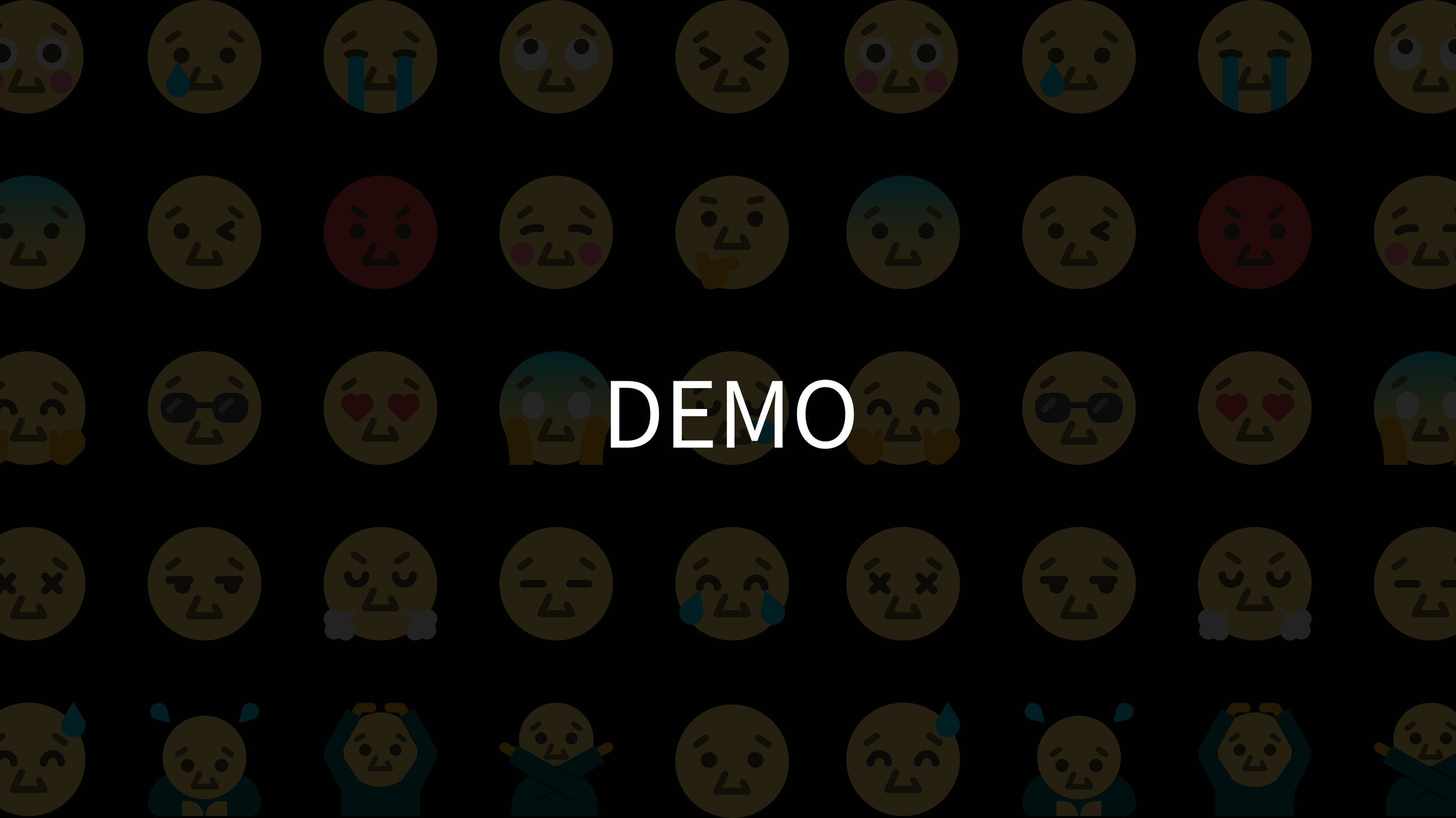
Vue.js入門 #06：配列の内容を繰り返し表示する **- v-for**

Vue.js入門 #07：**Vue.js**の動きにアニメーションを追加...

Vue.js入門 #08：コンポーネントで再利用可能なパーツ...

まとめ

- ウェブサイトでは必須ではない
- 何ができるのかの知識は持っておきたい
- コンポーネント単位での機能向き
- 他のフレームワークよりも
  導入の敷居が低い


DEMO

ご清聴ありがとうございました