



# *Cloud9*で始める *Vue.js*入門

2019年3月22日



# アジェンダ

---

- ▶ Cloud9とVue.jsについて
- ▶ npmを使用したプロジェクト作成(Vue CLI3)
- ▶ Vue基礎構文の理解(バインディング,コンポーネント,etc...)
- ▶ サンプルアプリケーションの制作



# Vue.js Cloud9とは

---



JavaScript3大フレームワークのひとつ、Angular/Reactより後発の為両者の良い所を採用し、現在JavaScriptフレームワークとして急成長中であり人気を集めている。

<< 3つの特徴 >>

- ・ディレクティブ
- ・データバインド
- ・学習コスト

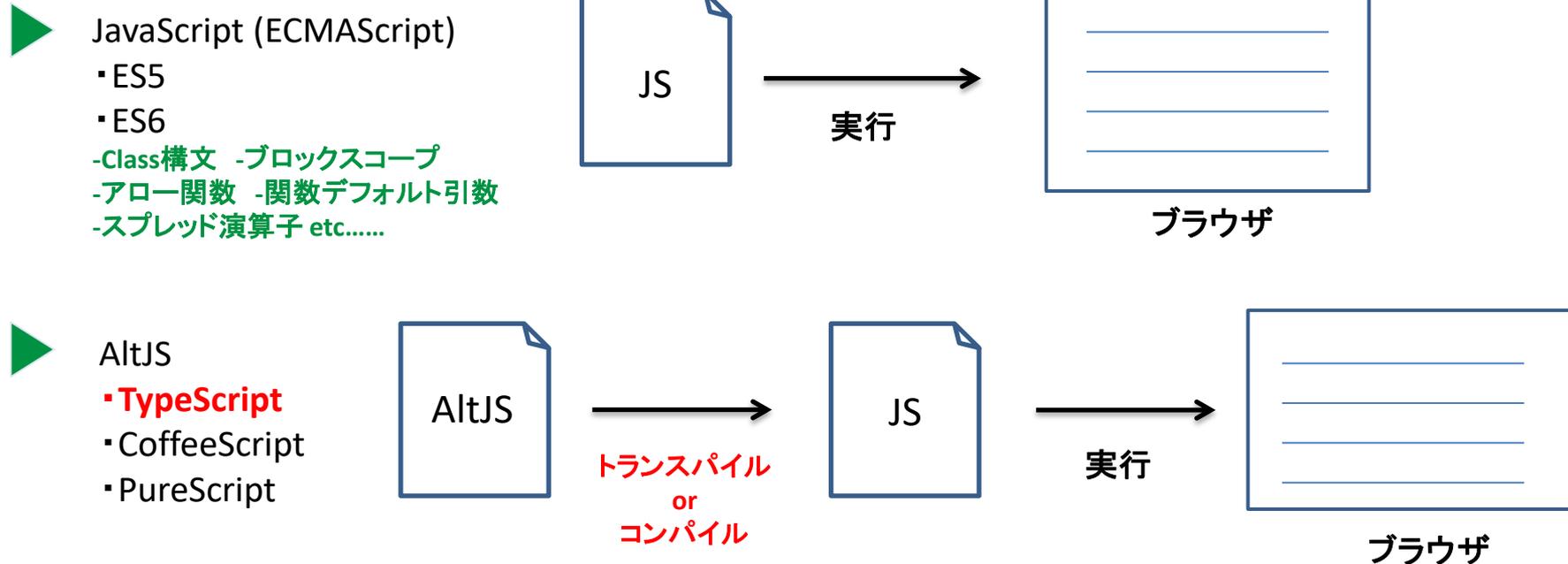


ブラウザIDE(統合開発環境)として登場し、現在はAmazonによって買収されAWS Cloud9サービスとして開発環境を提供している。

開発環境はIDEを主流としてきたが、時代と共に拡張性の優れたテキストエディタが登場し使用率が上昇する中、Cloud9はテキストエディタに負けず劣らずのインストール不要かつブラウザのみで開発を始める事が出来る非常に優れたサービス。



# JavaScript



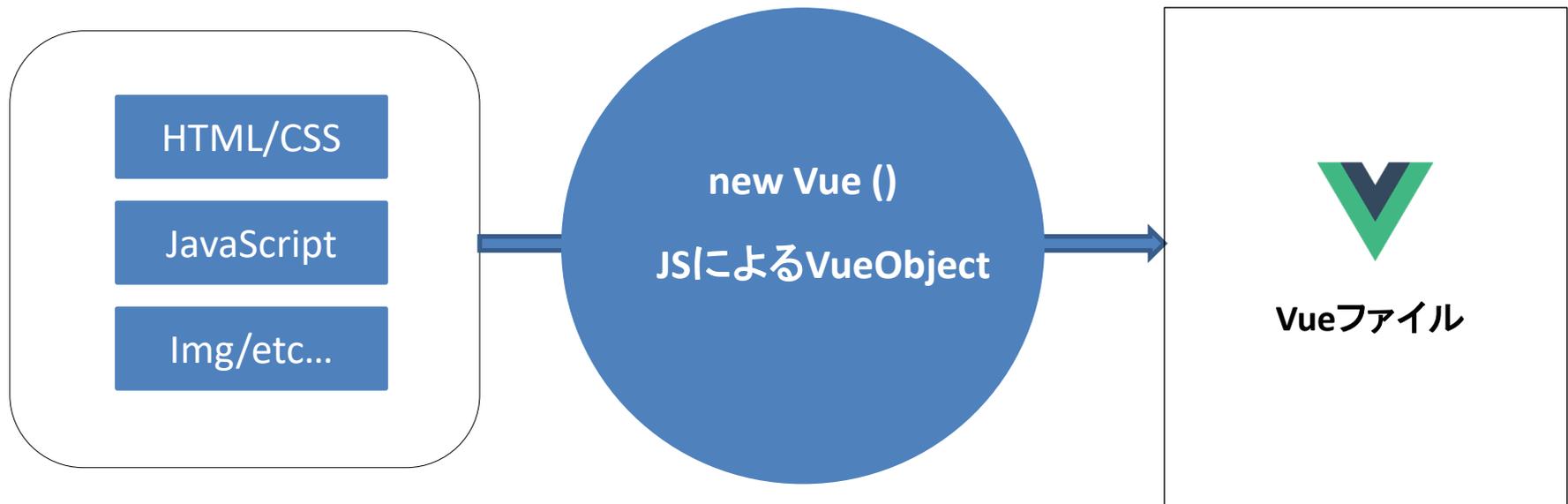
## Vue.js

JavaScript構文は基本的にES6基準での記述を行う事になります。  
他のライブラリを利用する事で、AltJSでの記述も可能、またVue.jsは今後TypeScriptを標準サポートするという話もあり、JavaScriptを学習してゆくのであればES6又はTypeScriptを学習する事でVue.jsを含めた他のフロントエンド領域に対応できます。



# Vue CLI (コマンドラインインターフェース)

- ▶ VueCLIによるプロジェクト作成と開発 (単一コンポーネント)
- ▶ Scriptタグによる読み込み or CDN (コンテンツデリバリーネットワーク)



**単一コンポーネント**  
**※1つの部品として利用**



# Vue.js の仕組み

<template></template>

HTML

```
<template>
  <div id="app">
    <header>
      <p id="title">Study Chat Room</p>
      <!-- ログイン -->
      <div class="login_box" v-if="user.uid" key="login">
        <div v-if="user.uid" id="user_icon">ログアウト</but
      </div>
      <!-- 未ログイン -->
      <div class="login_box" v-else key="logout"><input type="button" value="ログイン" /></div>
    </header>

    <!-- Firebase から取得したリストを描画 (トランジション付き) -->
    <transition-group name="chat" tag="div" class="list_content">
      <section v-for="{ key, name, image, message } in chat" :key="key">
        <div class="item-image"></div>
        <div class="item-detail"><span>{{ message }}</span></div>
      </section>
    </transition-group>

    <!-- 入力フォーム -->
    <form action="" @submit.prevent="doSend" class="form">
      <textarea
        v-model="input"
        :disabled="!user.uid"
        @keydown.enter.exact.prevent="doSend"></textarea>
      <button type="submit" :disabled="!user.uid" class="send-button">送信</button>
    </form>
  </div>
</template>
```

<script></script>

JavaScript

```
<script>
  // firebase モジュール
  import firebase from 'firebase/app'
  import 'firebase/auth'
  import 'firebase/database'
  // 改行を <br> タグに変換するモジュール
  import Nl2br from 'vue-nl2br'

  export default {
    components: { Nl2br },
    data () {
      return {
        // ユーザー情報
        user: {},
        // 取得したメッセージを入れる配列
        chat: [],
        // 入力したメッセージ
        input: '',
        // ユーザーアイコン
        user_icon: ''
      }
    },
    created () {
      // ...
    },
    methods: {
      // ...
    }
  }
</script>
```

<style></style>

CSS

```
<style scoped>
  header {
    padding: 10px;
  }
  form {
    margin-bottom: 10px;
  }
  section {
    margin-bottom: 10px;
  }

  #app {
    padding: 10px;
  }
  #title {
    font-size: 1.2em;
  }
  #user_icon {
    width: 40px;
    height: 40px;
  }
  .login_box {
    padding: 5px;
  }
  .list_content, form {
    margin-bottom: 10px;
  }
  .send-button {
    padding: 5px 10px;
  }
  .item-message div {
    padding: 5px;
  }
  .item-message div:before {
    padding-right: 5px;
  }
  .item-image {
    width: 60px;
    height: 60px;
  }
  .item-detail {
    padding: 5px;
  }
</style>
```

例) Chatアプリケーション



# プロジェクト作成手順

---

- ▶ **NVM更新** ..... NodeVersionを管理するツール
- ▶ **node更新** ..... Node.js更新により、npmを使用可能にする
- ▶ **npm更新** ..... ライブラリのパッケージを管理する為のツール  
npmによりVueCLIを管理する
- ▶ **Vue CLIインストール** Vue CLIによるVueプロジェクトの作成から依存関係の設定など、Vue.jsを使用した開発環境を構築する
- ▶ **Vueプロジェクト作成** Hellow Vue App



# プロジェクト作成手順

## ▶ NVM更新手順

- nvm ls
- nvm ls-remote
- nvm install v10.14.1
- nvm use v10.14.1
- nvm alias default v10.14.1

デフォルトがv10.14.1に変更されている  
事が確認出来たら完了

```
main:~/environment $ nvm ls
v6.15.0
-> v10.14.1
system
default -> v10.14.1
node -> stable (-> v10.14.1) (default)
stable -> 10.14 (-> v10.14.1) (default)
iojs -> N/A (default)
lts/* -> lts/argon (-> N/A)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.16.0 (-> N/A)
lts/carbon -> v8.15.0 (-> N/A)
lts/dubnium -> v10.15.1 (-> N/A)
main:~/environment $
```

※NVM,Node,npm,yarnなどのコマンドについては本講義では詳しい  
解説は行いませんが、フロントエンド領域を突き詰めて行く中で  
必須となるので興味のある方は個々に理解を深めてください。



# プロジェクト作成手順

---

## ▶ node更新

- `node -v`

※NMVによる設定値: v10.14.1となっていればOK

## ▶ npm更新

- `npm -v`
- `npm update -g npm`
- `npm -v`

※6.4.1となっていればOK  
場合によってはアップデート有



# プロジェクト作成手順

## ▶ Vue CLIインストール

- `npm install -g @vue/cli-init`

Vue CLIの最新版はVer 3だが、Cloud9設定などを含めて資料が多いVer 2方式を使用する。

## ▶ Vueプロジェクト作成

- `vue init webpack my-app` (プロジェクト名)
- `cd my-app npm`

```
main:~/environment $ vue init webpack my-app
? Project name my-app
? Project description A Vue.js project
? Author
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? Yes
? Pick an ESLint preset Standard
? Set up unit tests Yes
? Pick a test runner jest
? Setup e2e tests with Nightwatch? Yes
? Should we run `npm install` for you after the project has been created? (recommended) npm

vue-cli - Generated "my-app".

# Installing project dependencies ...
# =====
```

プロジェクト作成時、いくつかの選択肢を聞かれるが全てEnter押下によるデフォルト選択でOK

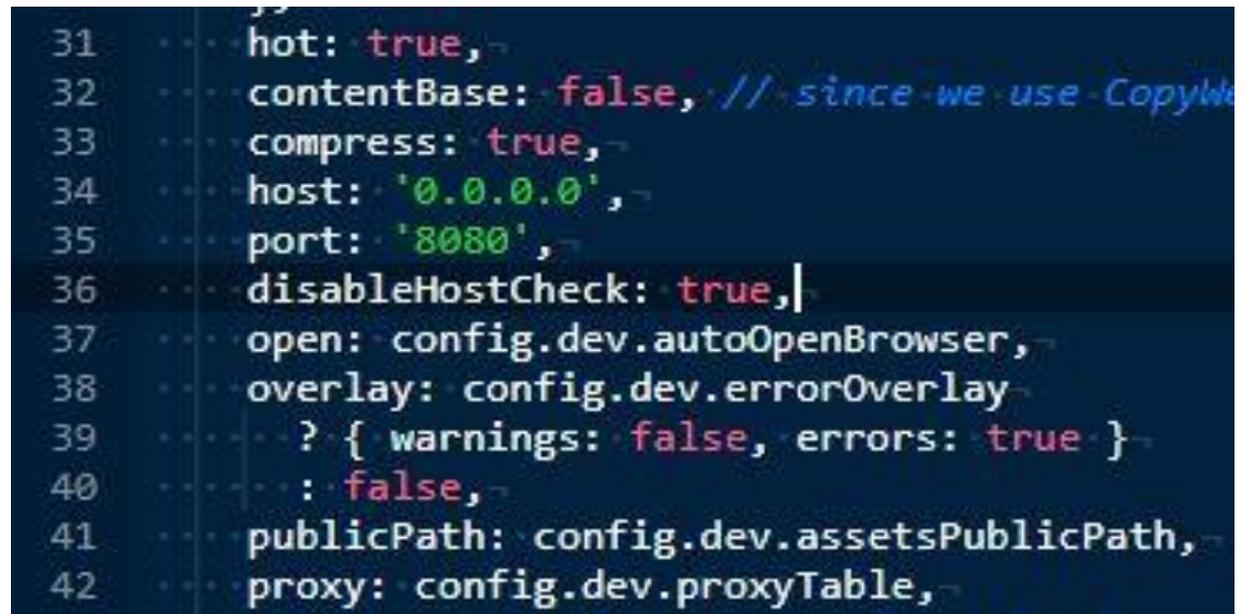
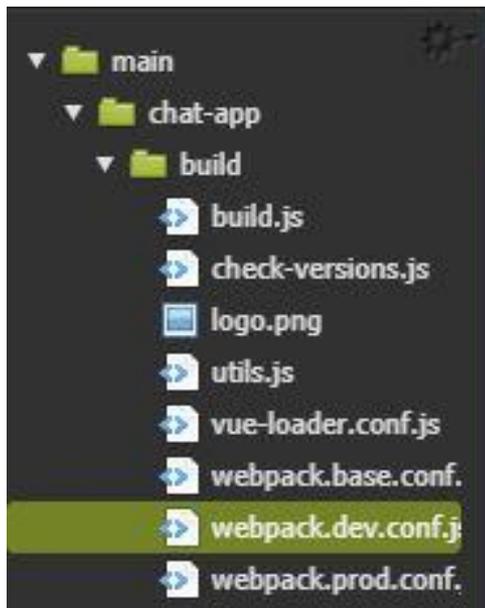
Vue CLIによるプロジェクト作成時のライブラリインストールや依存関係の設定など、開発環境を構築する際の設定を自動で行ってくれる



# プロジェクト作成手順

## ▶ Cloud9の設定

Cloud9を使用しない場合であれば、設定は不要。  
AWS Cloud9ローカル環境とnodeで起動するローカル環境の向き先が違う為  
Cloud9ではHost,Port設定を行う必要がある。





# Vueプロジェクト構造

## my-app

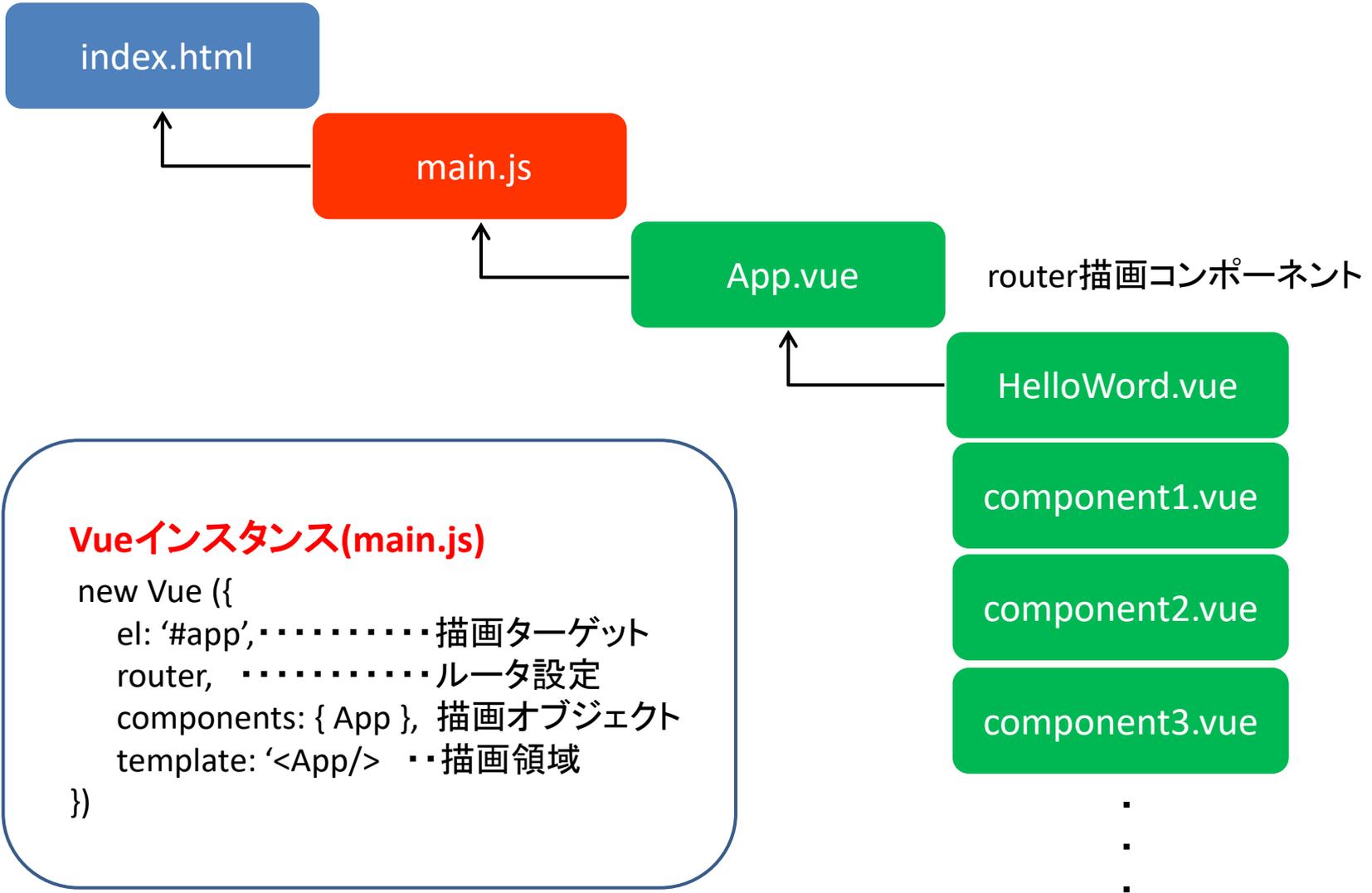
- └ build ..... build設定
- config ..... 設定全般
- dist ..... リリースファイル
- node\_modules ..... nodeモジュール
- src** ..... ソースファイル
- static ..... 静的リソース
- test ..... テストファイル
- Index.html** ..... index.html
- package-lock.json
- package.json ..... npmパッケージ管理jsonファイル
- README.md ..... 添付文書

## src

- └ assets ..... リソース管理
- components** ..... コンポーネント
- router ..... router設定
- App.vue** ..... Vueファイル
- main.js ..... **プロジェクトのメインファイル**



# Vue依存関係





# サンプルアプリケーション仕様

---

- ▶ 5×5マスの中にある1つの当たりを見つける
- ▶ 開放したマスに色を付ける
- ▶ 当たりマスには別の色付ける
- ▶ 当たりマス(答えは毎回変化)

## <使用する技術>

- ・ディレクティブ .....v-for v-on
- ・データバインド .....単方向バインディング
- ・ライフサイクルメソッド.....created



# Vue.js ディレクティブ

Vueインスタンス内に記述されたディレクティブ属性は、属性値の式に応じたDOM動作を行う。(V-XXX) による決まった書式で指定する。

## 【簡単な理解】

HTMLで例えるとタグの様な物、指定された属性を与える事で処理 or 動きをする

v-on:click='メソッド'

クリックイベントやform送信など、イベントの発火処理などに使用される。

v-class='クラス評価式 : クラス名'

評価式に応じてClassの追加・削除が行われる。

v-style='スタイル評価式 : スタイル名'

評価式に応じてStyleの追加・削除が行われる。

v-text, v-html, v-attr  
v-model, v-repeat

その他ディレクティブ属性

v-show='評価式'

評価式に応じてDOM要素のdisplayプロパティを操作

v-if='評価式'

評価式に応じてDOM要素の挿入・削除を行う。



# Vue.js データバインド

## ▶ 単方向バインディング

<template>側から<script>データオブジェクトへの参照のみを行う一方通行のバインディング

<template>【親】</template>



参照

<script>【子】</script>

## ▶ 双方向バインディング

<template>側からの参照のみではなく、データオブジェクトのデータ変更を可能にした双方通行のバインディング

<template>【親】</template>



参照 & 変更



<script>【子】</script>

## ▶ ワンタイムバインディング

単方向バインディングの実行が読み込みの1度のみ実行されるバインディング

<template>【親】</template>



参照(1回)

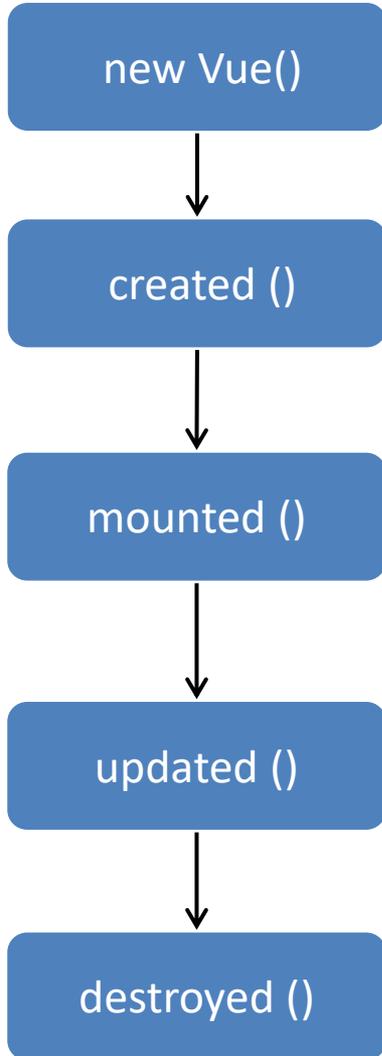
<script>【子】</script>

※参照方法

マスタッシュ構文{{値}}によりscript変数を参照



# Vue.js ライフサイクル



Vue.jsではhtmlファイルとしてのテンプレートファイルDOM(Document Object Model)を生成する過程で仮想DOMを生成し、最終的にDOMとして出力される。

仮想DOM生成のライフサイクルは左記の順番となるこの4つのフックメソッドは、仮想DOM操作の過程で頻繁に登場し使用頻度が高い為Vue.jsに触れながら理解を深めてください。

- Created** .....Vueインスタンス生成に1度呼ばれる  
初期化処理などはこちら
- Mounted** ..... VueインスタンスがDOMにマウントされる時1度呼ばれる
- Update** ..... データが更新されDOMに反映された時1度呼ばれる
- Destroyed** ..... Vueインスタンスが削除された時よばれる

※各フックメソッドにはBeforeメソッドが存在する  
各フックメソッドが呼ばれる前に呼ばれるメソッド



# 学習コスト

---



## シンプルな構成

<template><script><style>3つの領域によって明確な区切りがありとてもシンプルな作り  
各コンポーネントは完結した1つの部品となりローカルスコープを持つ為、データの競合が起こらない。



## 公式ドキュメントが日本語対応

日本語ドキュメントが充実しており、React,Angularと比べてとても親切な説明が多い  
ドキュメントサイト自体が読みやすいのも大きなメリット。



## ※コードが読みやすい※

- シンプルな構成により実装すべき項目や、実装箇所が一目で理解する事が出来る
- html/css知識があればJavaScript知識のみで実装が可能であり、先入観なく開発が行える。
- 目に優しい(個人差有り)



# V-for

- ▶ **V-for** item in itemsの形で指定する。  
JavaScriptで言うforEach文と同じ動きをします。  
【 V-for='X in Y' key='Y.id' 】  
※keyはv-forを識別する為に必要となります。  
ループ内では{{X.id}} {{X.name}}で各要素を参照

## **配列Object**

```
Y=[  
  {'id':1, 'name':hoge},  
  {'id':2, 'name':huga},  
];
```

## ▶ **Script記述**

- data      コンポーネント内の変数を記述(変数定義)
- created   ライフサイクルメソッド(初期化処理)
- methods   処理の記述(メソッド)



# ご静聴ありがとうございました。

---

- ▶ Vue公式ドキュメント <https://jp.vuejs.org/>
- ▶ VueRouterドキュメント <https://router.vuejs.org/ja/>
- ▶ Vuexドキュメント <https://vuex.vuejs.org/ja/>

本講座ではVue.jsの基礎構文の説明でしたが、Vue.jsは開発をスムーズにする多くの仕様やライブラリが存在します。  
今後Vue.jsの学習を進めて行く方は、Vue Router・Vuex・Nuxt.jsなどをお勧めします。