

目錄

[安裝環境 2](#_Toc61487524)

[軟體 2](#_Toc61487525)

[VSCODE套件 2](#_Toc61487526)

[VUE 2](#_Toc61487527)

[VUE2 2](#_Toc61487528)

[Preview on Web server 2](#_Toc61487529)

[第一個HelloWorld 3](#_Toc61487530)

[引用JS檔 3](#_Toc61487531)

[雙向綁定 3](#_Toc61487532)

[處理用戶端輸入v-on:click 4](#_Toc61487533)

[元素的邏輯判斷與顯示v-if&v-show 4](#_Toc61487534)

[用V-on:事件 7](#_Toc61487535)

[v-on:click簡寫@click 7](#_Toc61487536)

[相關影片 7](#_Toc61487537)

[Vue 開發環境介紹 7](#_Toc61487538)

[Vue.js 教學 - 幼幼班入門篇 (上) 7](#_Toc61487539)

# 安裝環境

## 軟體

VSCODE

NODE.JS

VUE

## VSCODE套件

### VUE

Hilight .Vue檔的套件

### VUE2

### Preview on Web server

* Preview on side panel (ctrl+shift+v):
* Stop the web server (ctrl+shift+s):
* show UI Page (ctrl+shift+u):

## 第一個HelloWorld

|  |
| --- |
| <html>  <head></head>  <body>  <div id ="app">      <p>{{message}}</p>  </div>  <script src="https://cdnjs.cloudflare.com/ajax/libs/vue/1.0.26/vue.min.js"></script>  <script>      new Vue({          el:'#app',          data:{          message:'hello world!!'          }      });      </script>  </body>  </html> |

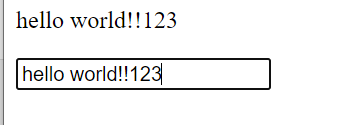
### 引用JS檔

在<body>引用VUE的JS檔

|  |
| --- |
| <body>  <script src="https://cdnjs.cloudflare.com/ajax/libs/vue/1.0.26/vue.min.js"></script>  </body> |

### 雙向綁定

|  |  |  |
| --- | --- | --- |
| |  | | --- | | <div id="app">  <p>{{ message }}</p>  <input v-model="message">  </div> |  |  | | --- | |  | |
| new Vue({  el: '#app',  data: {  message: 'Hello Vue.js!'  }  }) |

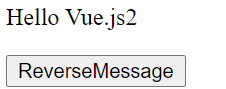
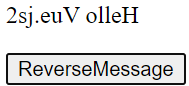


### 處理用戶端輸入v-on:click

在button加上”v-on:Lclick”

呼叫reverseMessage的函數

|  |
| --- |
| <div id ="app2">      <p>{{message2}}</p>      <button v-on:click="reverseMessage">ReverseMessage</button>  </div> |
| var app2 = new Vue({          el:'#app2',          data:{              message2: 'Hello Vue.js2'          },          methods:{              reverseMessage: function () {                  this.message2 = this.message2.split('').reverse().join('')              }          }      }); |

### 元素的邏輯判斷與顯示v-if&v-show

#### V-if

|  |
| --- |
| <!--單純判斷布林-->  <p v-if="loading">ERROR</p>  <!—數學邏輯判斷-->  <p v-if="score >= 60">分數:{{score}},及格</p>  <p v-else="score < 60">分數:{{score}},不及格</p> |

#### v-show

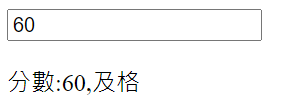
|  |
| --- |
| <!--單純判斷布林-->  <p v-show="loading">ERROR2</p>  <!—數學邏輯判斷-->  <p v-show="score < 60">分數:{{score}},不及格</p> |

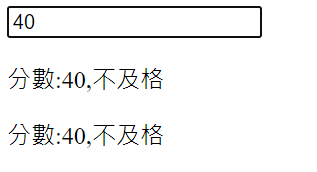
#### v-if&v-show範例

|  |
| --- |
| <div id ="app">      <input v-model="score"></input>      <p v-if="score >= 60">分數:{{score}},及格</p>      <p v-else="score < 60">分數:{{score}},不及格</p>      <p v-show="score < 60">分數:{{score}},不及格</p>  </div> |
| var app = new Vue({          el:'#app',          data:{              message: 'Hello Vue.js',              score :60,              loading:false          },          methods:{              reverseMessage: function () {                  this.message2 = this.message2.split('').reverse().join('')              }          }      }); |

判斷loading=false，只會顯示Success

* 用v-if不會產生元素
* 用v-show會產生元素，只是加上了”style=”display:none;””的屬性





### 調用函數method

然而許多事件處理邏輯會更為複雜，所以直接把 JavaScript 代碼寫在 v-on 指令中是不可行的。因此 v-on 還可以接收一個需要調用的方法名稱。

示例：

<div id="example-2">

<!-- `greet` 是在下面定義的方法名 -->

<button v-on:click="greet">Greet</button>

</div>

var example2 = new Vue({

el: '#example-2',

data: {

name: 'Vue.js'

},

// 在 `methods` 物件中定義方法

methods: {

greet: function (event) {

// `this` 在方法裡指向當前 Vue 實例

alert('Hello ' + this.name + '!')

// `event` 是原生 DOM 事件

if (event) {

alert(event.target.tagName)

}

}

}

})

// 也可以用 JavaScript 直接调用方法

example2.greet() // => 'Hello Vue.js!'

#### 內聯處理器中的方法

除了直接綁定到一個方法，也可以在內聯 JavaScript 語句中調用方法：

<div id="example-3">

<button v-on:click="say('hi')">Say hi</button>

<button v-on:click="say('what')">Say what</button>

</div>

new Vue({

el: '#example-3',

methods: {

say: function (message) {

alert(message)

}

}

})

#### 計算屬性Computed

根據VUE的物件變動後會重新計算呈現(要加上this)

以下是price/count是被前端按鈕綁定，一旦更新就會跟著變動

|  |
| --- |
| <button :disabled="!who || what.price\*count<=0">加入${{getsum}} </button> |
| var app = new Vue({        data:()=>({          a:1,          title: '下午茶訂單系統',          items: ITEM,          order: [{who: '阿嘎', what: ITEM[0], count: 1},                  {who: '阿嘎', what: ITEM[1], count: 2},                  {who: '阿嘎', what: ITEM[3], count: 3}],          who: undefined,          what: ITEM[0],          count: 1,          sum: 0        }),        method:{          addoder:function(){            this.order.push({who: '阿嘎', what: {name: '真奶', price: 100}, count: 2});          }        },        computed:{          getsum:function(){            this.sum = this.what.price\*this.count;            return this.sum;          }        }    })    app.$mount('#app'); |

## 用V-on:事件

### v-on:click簡寫@click

@change

V:on

# 相關影片

## Vue 開發環境介紹

<https://courses.hexschool.com/courses/vue/lectures/11948972>

## Vue.js 教學 - 幼幼班入門篇 (上)

https://www.youtube.com/watch?v=8O3teHziU\_E