

VueJS

Lecture1

Goal

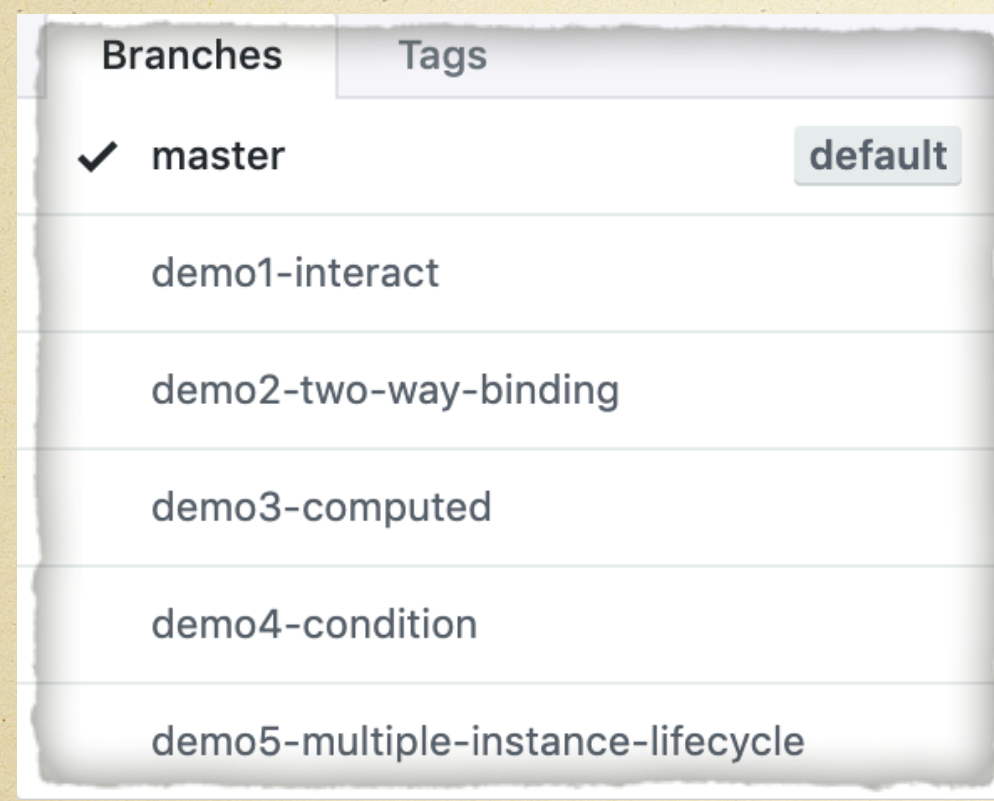
- 建構Vue的基礎知識
- 理解講座範例
- 有能力建構一個Vue Application

Agenda

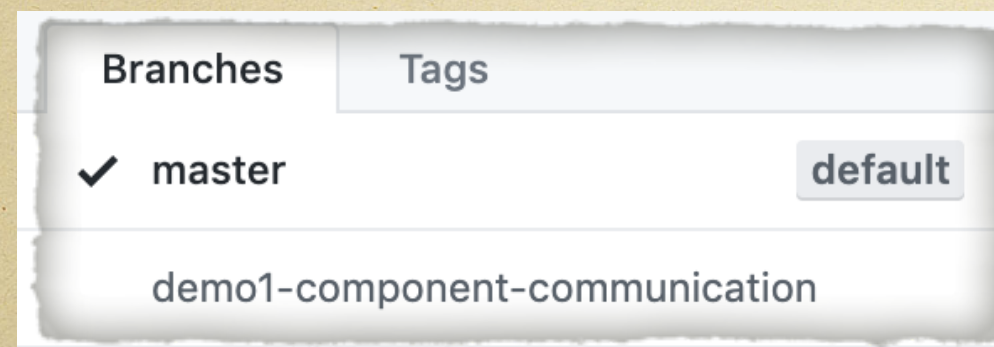
- 使用VueJS控制DOM
- 資料處理（條件控制及列表呈現）
- 探討Vue實例（Vue Instance）
- 認識組件（Component）
- 建立VueJS開發環境（Modern frontend workflow）
- 運動時間

《資源下載-講座範例》

➤ vue-lecture1講座範例-Demo1 - Demo5

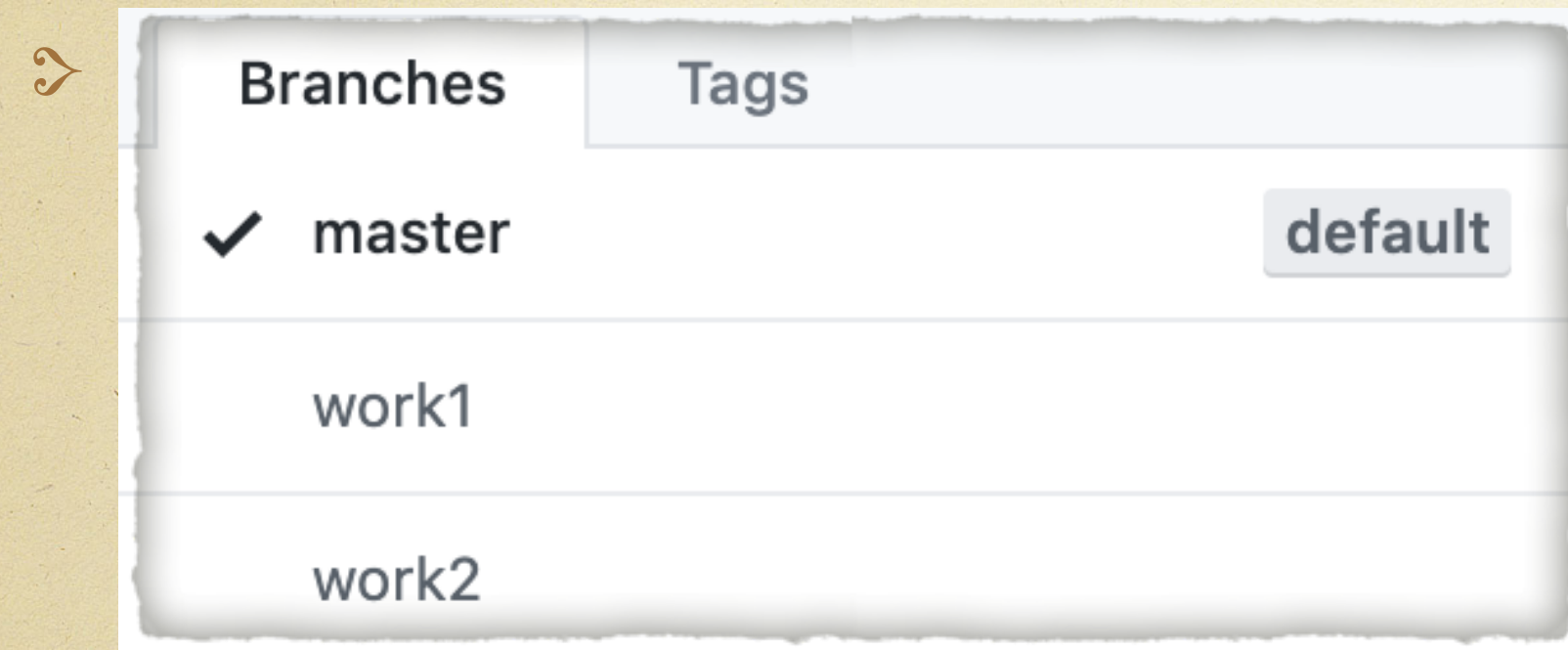


➤ vue-workflow講座範例-Demo1

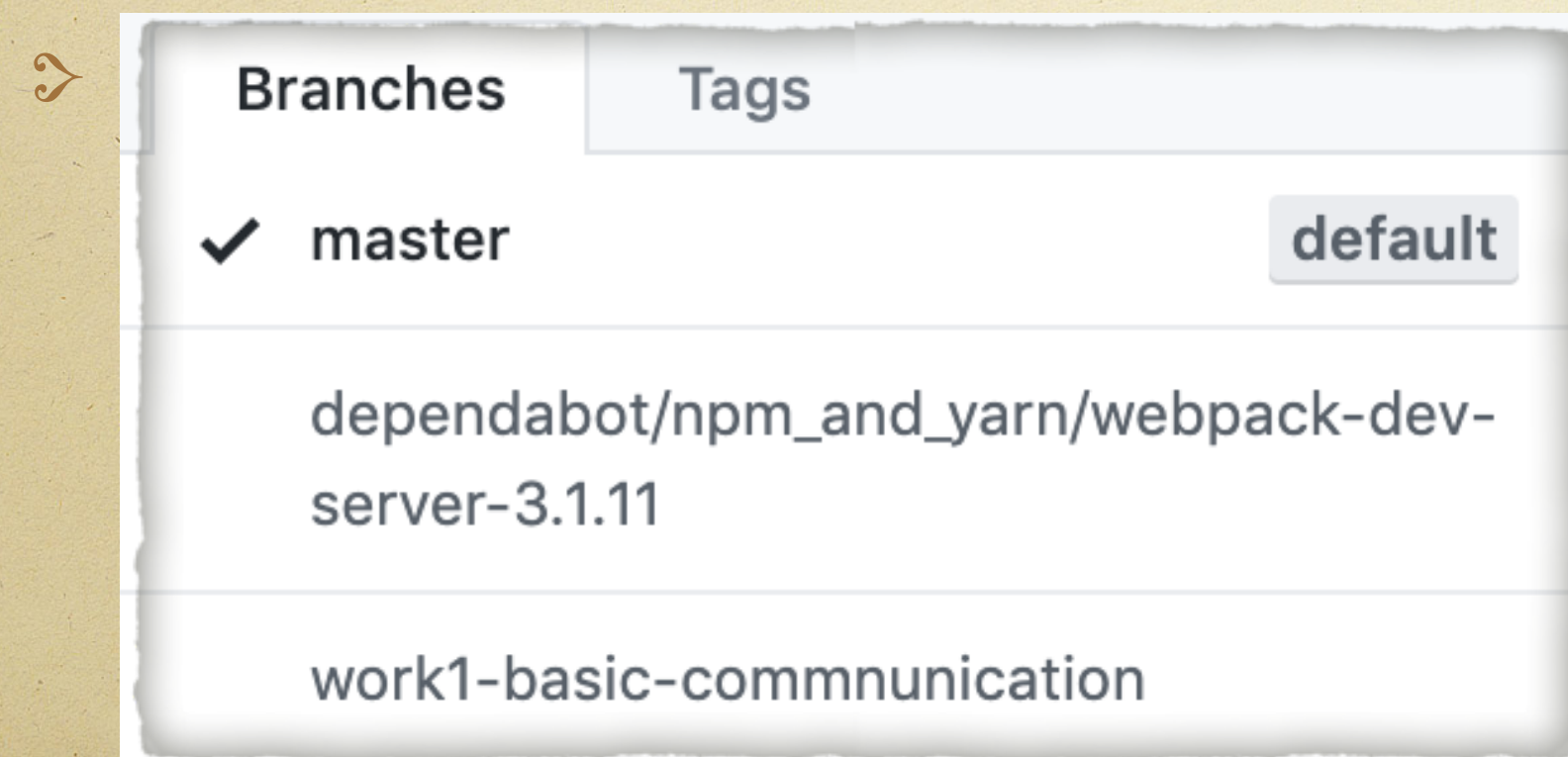


《資源下載-練習範本》

➤ vue-blank練習範本-Work1 - Work2



➤ vue-workflow-blank練習範例-Work1



《使用VueJS控制DOM》

認識模板 (Template Syntax)

- VueJS是基於DOM作為Template
- 使用“Mustache”語法將數據綁定至DOM呈現
- 被綁定之數據並非HTML，而是被視為純文字 (Plain Text)
 - 優勢：可防止XSS攻擊
- 數據可綁定至HTML Attribute，需使用指令 (Directive)
 - 指令 (Directive) 為帶有“v-”前綴的特殊屬性
 - 參考範例：v-bind、v-on、v-if、v-for、v-show
 - [:] 常用說明1：v-bind:<參數屬性>=“綁定的值或變數”
 - [@] 常用說明2：v-on:<監聽事件>=“觸發事件執行的行為”
- 事件綁定修飾符 (Modifier) 範例
 - <form v-on:submit.prevent=“doSomething”></form>
- 模板中所有數據綁定皆可支援單個Javascript表達式

《練習時間》

《使用VueJS控制DOM》

表單輸入綁定 (Two Way Binding)

- 在表單元件中可利用v-model directive執行數據雙向綁定
- 兩個步驟實現數據雙向綁定
 - 監聽事件
 - 進行數據綁定
- 只針對Vue實例中的data進行綁定

《使用VueJS控制DOM》

Computed & Method

- Computed屬性，使用方式可以如同data object（無需使用括號）
- Computed屬性，指定依賴屬性發生變化時，才會重新更新執行
- Computed屬性會進行Cache，若依賴屬性並無變化，則返回前次結果
- Watch屬性用於監聽指定屬性的變化，並進行響應操作
- 兩者很類似，一般大多數情況，建議使用Computed屬性處理運算邏輯

《使用VueJS控制DOM》

Watch

- Watch屬性用於監聽指定屬性的變化，並進行響應操作
- 與Computed類似，一般大多數情況，建議使用Computed屬性處理運算邏輯

《資料處理（條件控制及列表呈現）》

條件式、迴圈、物件

➤ v-if

➤ v-show

➤ v-for

《練習時間》

《探討Vue實例 (Vue Instance) 》

建立並存取資訊

- 允許多個Vue實例
- 多個Vue實例得以互相存取資料
- 在外部皆可存取Vue實例中的資訊
- 無法於外部建立新的元素並指定給Vue實例
- Proxy僅限於Vue實例中初始化建構的data、method、etc...

《認識組件 (Component) 》

作用域 (Local & Global)

Global Registration

```
// Define a new component called button-counter
Vue.component('button-counter', {
  data: function () {
    return {
      count: 0
    }
  },
  template: '<button v-on:click="count++">You clicked me {{ count }} times.</button>'
})
```

Local Registration

```
new Vue({
  el: '#app',
  components: {
    'component-a': ComponentA,
    'component-b': ComponentB
  }
})
```

使用Vue Instance 註冊Component缺點

- Global 組件名稱不得重複
- 元件註冊預設只能使用 HTML 及 ES5 JavaScript，不支持CSS設定
- 字串模板的問題，分行、辨識度等等，開發效率不彰

《探討Vue實例 (Vue Instance) 》

生命週期 (點擊可查看官網圖解)

- 1. **beforeCreate** : Vue Instance 初始化資料和事件
- 2. **created** : Vue Instance 已建立
- 3. **beforeMount** : 執行編譯Template (或者是 el 的template) , 將其物件資訊指定好, 轉換成html code
- 4. **mounted** : 編譯後的template進行掛載, 寫入DOM
- 5. **beforeUpdate** : 當資料異動或是被呼叫, 資料已改變但還不會渲染 View 。
- 6. **updated** : 當資料更新完成, 則驅動 DOM 元素重新渲染 View 。
- 7. **beforeDestroy** : 執行銷毀指定Vue Instance之前。
- 8. **destroyed** : Vue Instance銷毀, 所有的DOM 元素皆移除綁定、移除監聽事件、etc ... 。

《建立VueJS開發環境 (Modern frontend workflow) 》

使用Vue CLI幫助我們建立開發環境

- Vue CLI 最主要的用途是提供VueJS Project Template，協助快速建立專案所需的安裝環境，其中Project Template如下可選擇
- 1. Simple
- 2. Webpack-Simple （範例使用這個）
- 3. Webpack
- 4. Browserify/Browserify-simple

《建立VueJS開發環境 (Modern frontend workflow) 》

安裝Vue CLI

➤ 需要安裝NodeJS

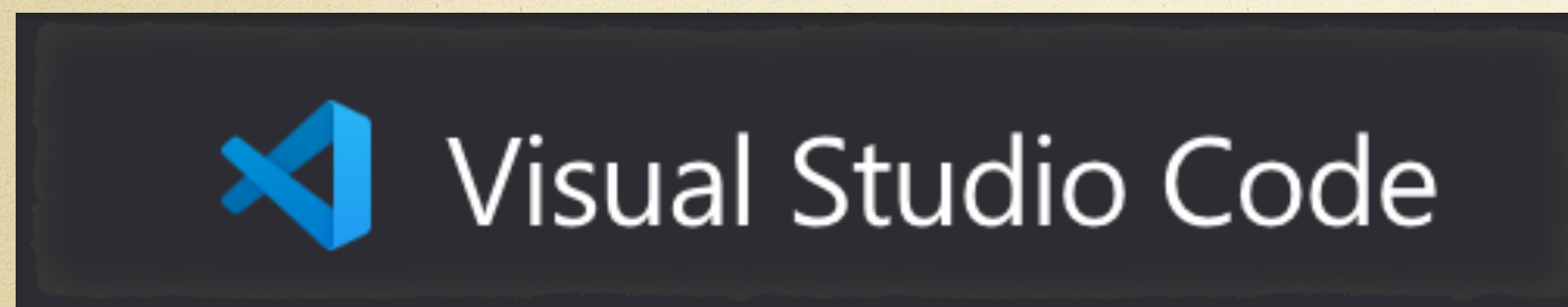


➤ 執行 `npm install -g vue-cli`

《建立VueJS開發環境（Modern frontend workflow）》

使用IDE及套件

- 通用平台考量，講座中使用VSCODE（點選連結至下載頁面）



- 安裝Plugin

Vetur



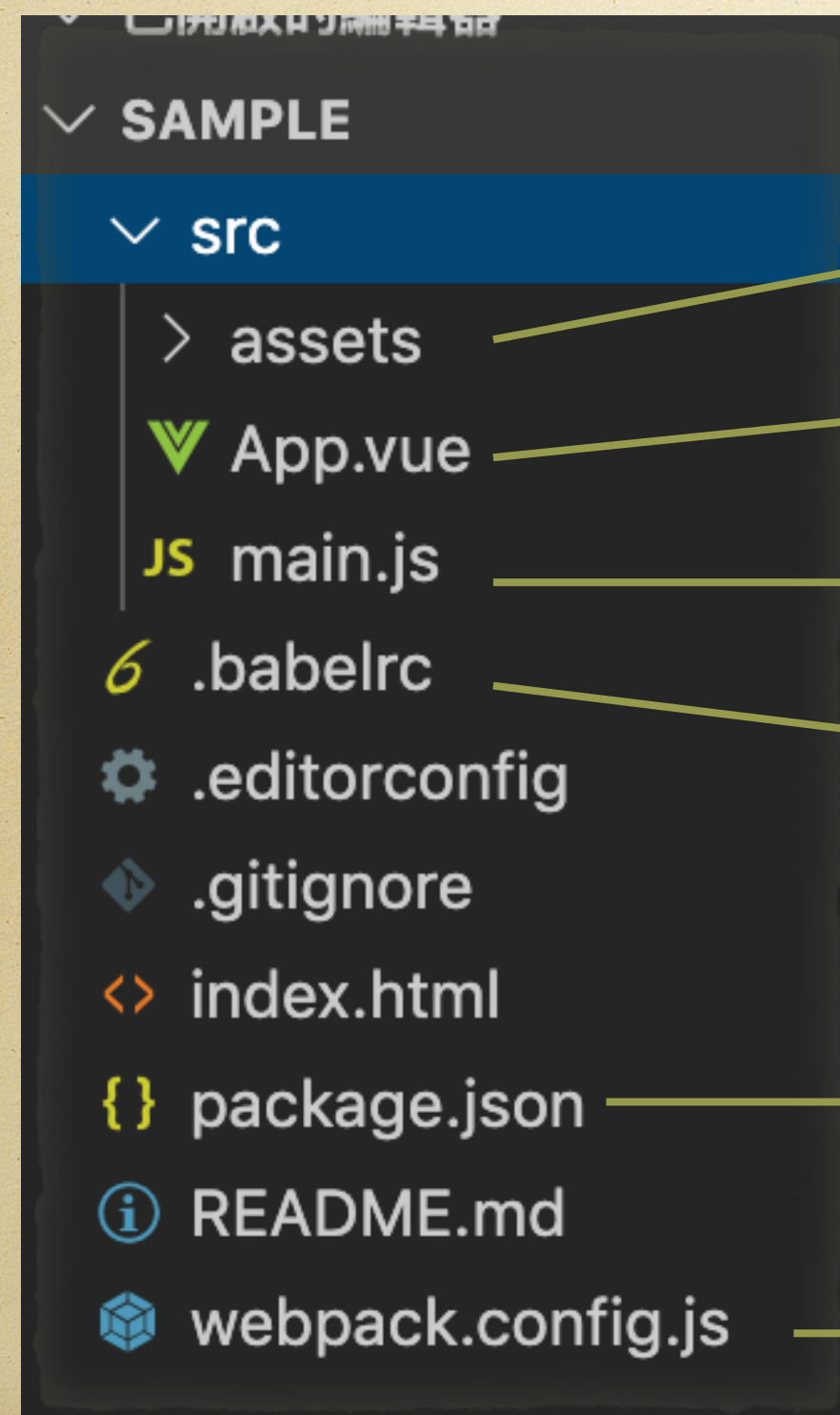
Auto Close Tag



《建立VueJS開發環境（Modern frontend workflow）》

使用Vue CLI建立專案

➤ `vue init <project-template> <project-name>`



放置靜態檔案

單一組件模板

程式入口點

JS編譯器

NPM管理套件

Webpack設定

```
vue init webpack-simple vue-sample
```

```
? Project name vue-sample
? Project description A Vue.js project
? Author Louis Chuang <louis.chuang@thinkpower-info.com>
? License MIT
? Use sass? (y/N) y
```

vue-cli · Generated "vue-sample".

To get started:

```
cd vue-sample
npm install
npm run dev
```


《認識組件（Component）》

單一組件模板Single-file-template

- 支援Webpack打包工具
- 可以解決template字串模板的問題
- 支援CSS

《認識組件（Component）》

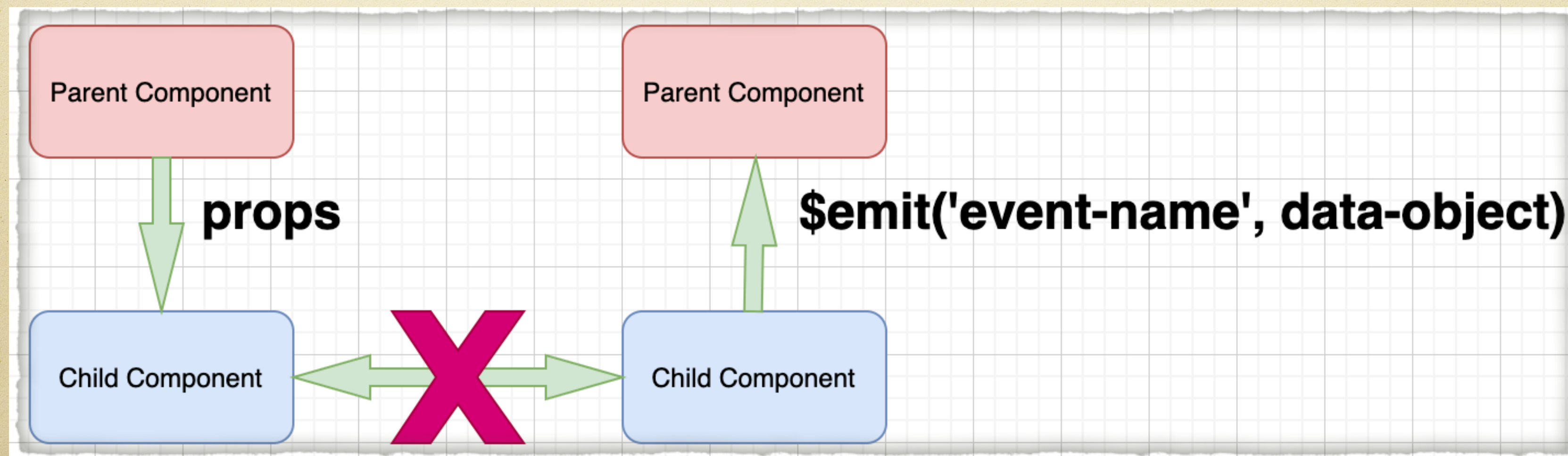
建立並開始使用組件

➤ 使用Single-file-template

《認識組件 (Component) 》

階層關係&資料傳遞

- 允許 Parent component » Child component : props
- 允許 Child component » Parent component : `$emit('eventName', data)`
- 不允許 Child component » Child component

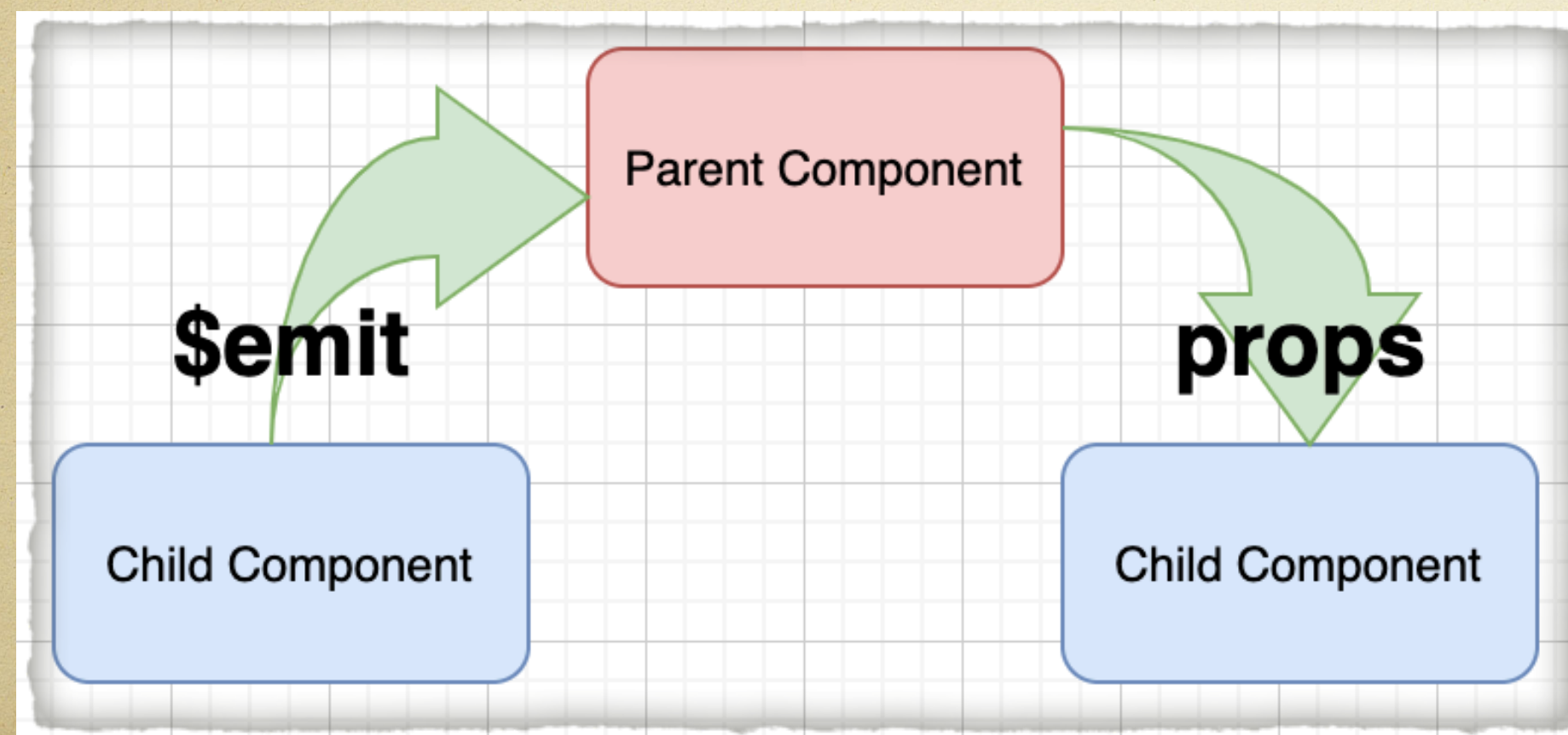


《認識組件 (Component) 》

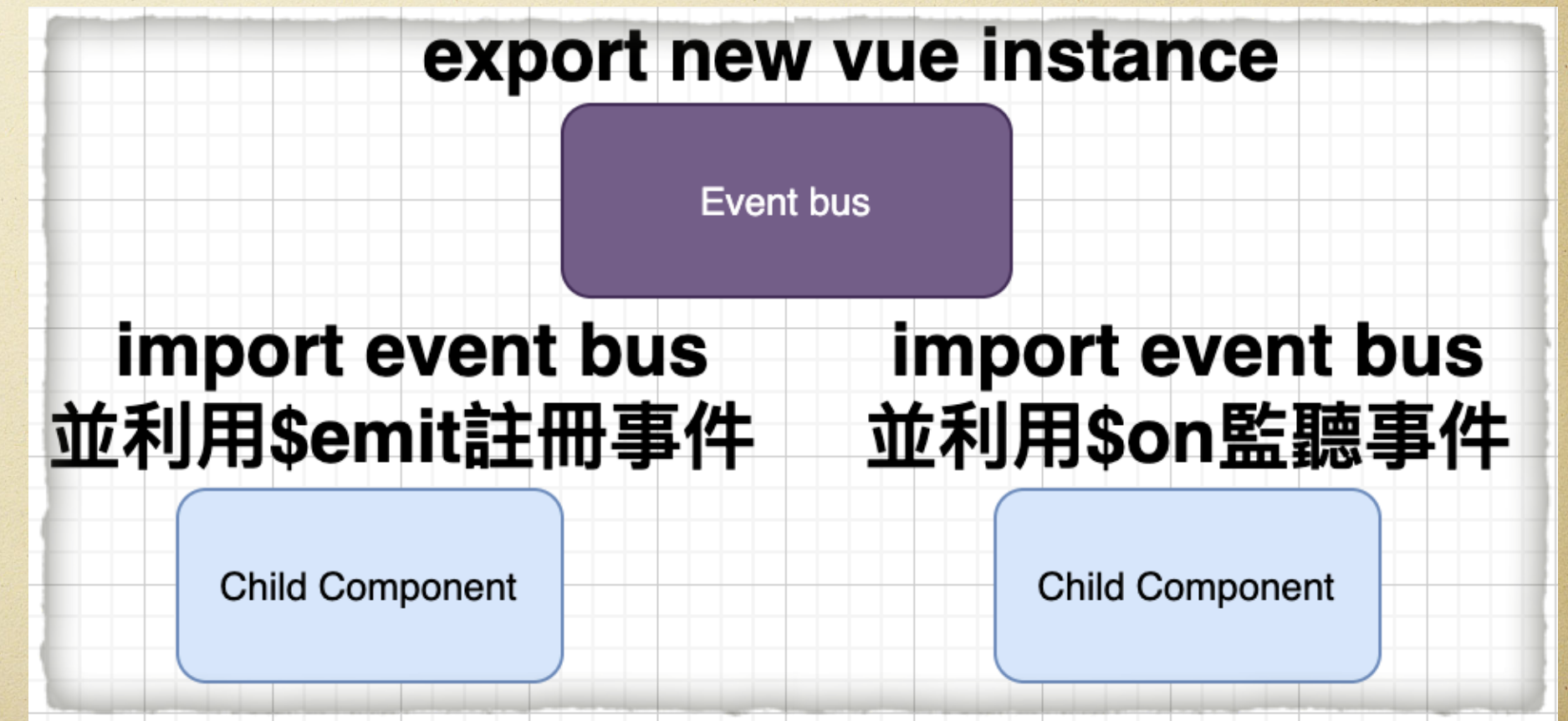
Child component » Child component

- Way1 : 經由向上傳遞，在更新至指定的Child Component : `$emit()` + `props`
- Way2 : `eventBus`: `$emit()` + `$on()`

Way1



Way2



《練習時間》

《運動時間》

需求&規則

- 請完成一個簡單的Todo List -> 下載練習範本
- 使用Workflow進行開發
- 必須使用多個組件完成此功能

《謝謝大家》

任何問題歡迎隨時提出