**Vue.js 透過new vue() 創建一個Vue實例 並且在裡面傳入包含屬性及方法的物件**

**El (element) 要綁定的DOM元素**

**Data 要綁定的資料內容**

**DOM元素：Document Object Model 中文是文件物件類型**

**同時會在名為APP 的應用程式data當中定義變數text**

**接下來使用 {{ }} 串接資料 把資料render畫面上**

**\*vue應用程式可以水平同時存在 但不可以有上下階層的關係**

**因為資料只會讀取最外面的data內容**

**範例：**



**聲明式渲染與命令式渲染**

**Vue.js 就是採用簡潔的模板 使用聲明式軒染將數據渲染上DOM元素**

**命令式渲染就是手動操作DOM元素 需要一步一步告訴程式碼該如何去做**

**，一個口令一個動作，使用前通常都要取得目標元素，再來把值放入元素中**

**用JQuery寫旁邊範例的寫法就是**

**範例:**

$("#app").text("這是我的第一個 Vue 應用程式")

**聲明式渲染 可以讓開發者更關注在狀態與資料表現，只要告訴程式碼想要什麼結果的感覺。因此vue透過el (element) 屬性決定執行的範圍 就會把應用程式data屬性存入的值傳送給 {{ text }} 模板 所以當資料改變時 HTML也會跟**

**著改變。**

**結論**

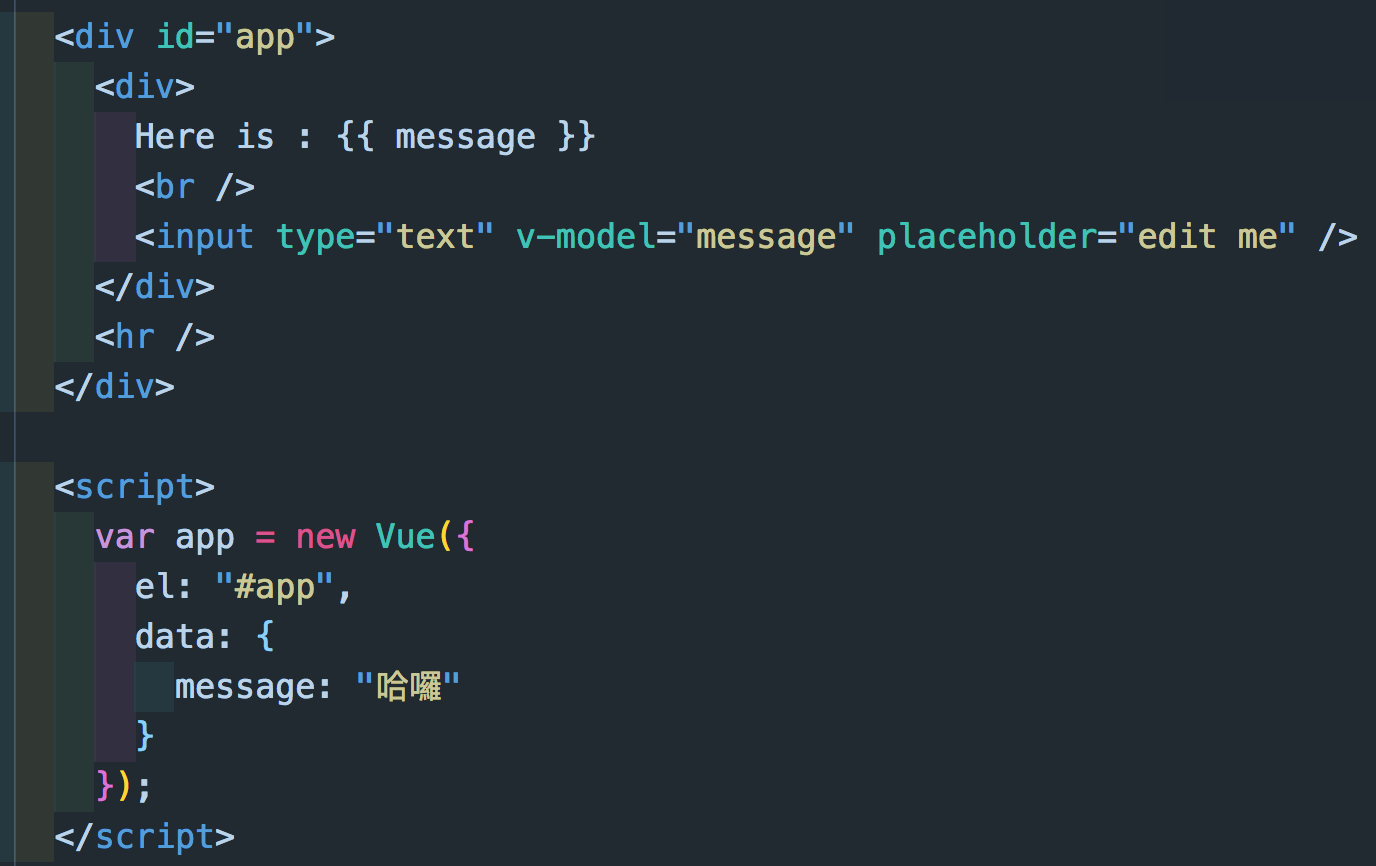
**當使用vue做開發套件時 只要專注於資料同時決定資料要渲染在哪個html標前的哪個位置即可，剩下的就交給vue去執行。**

**Vue的特色在於資料雙向綁定 (Two-way-Binding)**

**資料最常見的為v-model**

**v-model常用於表單及元素來做雙向數據綁定 綜合其他指令 v-bind與v-on一樣 ， 除了綁定與呈現資料外，還會監聽事件來做資料更新**

**範例:**



**被v-model綁定data為message的input標籤**

**同時在input輸入資料 {{message}}裡的資料也會同步更新**

**v-model也常用在checkbox的用法**

**像check box 勾選**



**上述就是透過v-model綁定checkbox1的狀態 以true與false切換勾選狀態**

**或是 checkbox的複選**

****

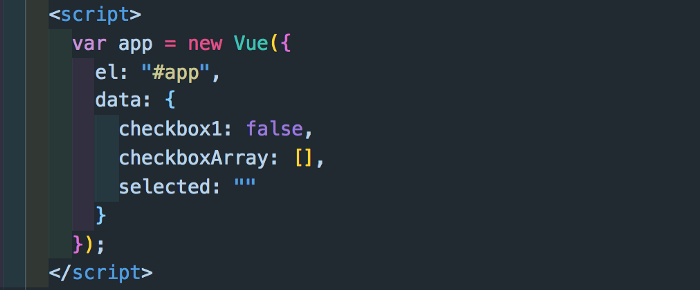
**透過v-model綁定checkboxArray 讓勾選的值跑進p裡的{{checkArray}} 來呈現畫面 告訴使用者今天晚餐有哪些料**

**還有select 下拉式選單**

****

**透過v-model綁定selected 勾選則加入下拉式選單內**

**上述三個應用程式當中的data分別有 checkbox1 checkboxArray selected**

**如下圖**

**v-model的修飾符**

**1.lazy v-model在每次input事件觸發時 將輸入框的值與數據進行同步 如添加lazy修飾符 轉變為使用change事件同步 即離開輸入欄位時才輸出資料**

**範例:**

**<input type=”text” v-model.lazy=”lazyMsg”>**

**2.number 如果想自動將用戶所輸入的內容轉換為數字型態 可以直接在v-model添加number修飾符**

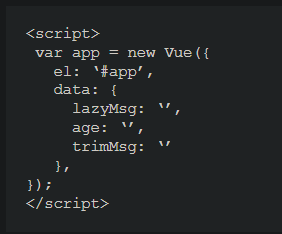
**範例:**

**<input type=”text” v-model.number=”age”>**

**3.trim 為避免用戶物輸入空白字元 因此可以透過v-model添加trim修飾符 自動過濾掉用或所輸入的首尾空白字符**

**<input type=”text” v-model.trim=”trimMsg”>**

**上述應用程式的data分別用lazyMsg age trimMsg**

****