

1. Vue 樣板的用途: 可以根據程式的資料產生最終的畫面:

## 樣板的用途

可以根據程式中的資料  
產生最終的畫面

組件名稱.vue

```
<script setup>
  let data="Hello";
</script>
<template>
  <div>[[ data ]]</div>
</template>
<style scoped>
  /* CSS 設定 */
</style>
```

## 2.0 : 今天二個主題 : if 判斷式 及 迴圈

2. if 判斷式, 使用 v-if = "布林值" 語法:

## if 判斷式

使用 v-if="布林值" 語法

組件名稱.vue

```
<script setup>
  let data=true;
  let number=3000;
</script>
<template>
  <div v-if="data">顯示</div>
  <div v-if="!data">不顯示</div>
  <div v-if="number>2000">大於 2000</div>
</template>
<style scoped>
  /* CSS 設定 */
</style>
```

3. 使用 if ... else if ... else 判斷式: 使用 v-else-if = "布林值"語法 ,必須搭配 v-if 使用 ,可搭配 v-else

## if ... else if ... else 判斷式

使用 v-else-if="布林值" 語法  
必須搭配 v-if 使用 , 可搭配 v-else

組件名稱.vue

```
<script setup>
  let number=15;
</script>
<template>
  <div v-if="number>20">大於 20</div>
  <div v-else-if="number>10">大於 10</div>
  <div v-else>小於等於 10</div>
</template>
<style scoped>
  /* CSS 設定 */
</style>
```

#### 4. for迴圈 - 陣列：使用 v-for="變數 in 陣列" 語法

## for 迴圈 - 陣列

使用 `v-for="變數 in 陣列"` 語法

組件名稱.vue

```
<script setup>
  let data=[1, 2, 3];
</script>
<template>
  <div v-for="n in data">Item {{ n }}</div>
</template>
<style scoped>
  /* CSS 設定 */
</style>
```

#### 5. for 迴圈 - 陣列索引：使用 v-for="(變數, 變數)" in 陣列的語法: 其中第二個變數為索引:

## for 迴圈 - 陣列索引

使用 `v-for="(變數, 變數) in 陣列"` 語法

組件名稱.vue

```
<script setup>
  let data=["A", "B", "C"];
</script>
<template>
  <div v-for="(text, index) in data">
    {{ index }} - {{ text }}
  </div>
</template>
<style scoped>
  /* CSS 設定 */
</style>
```

#### 6. for 迴圈 - 物件：使用 v-for="(變數, 變數) in 物件" 語法

## for 迴圈 - 物件

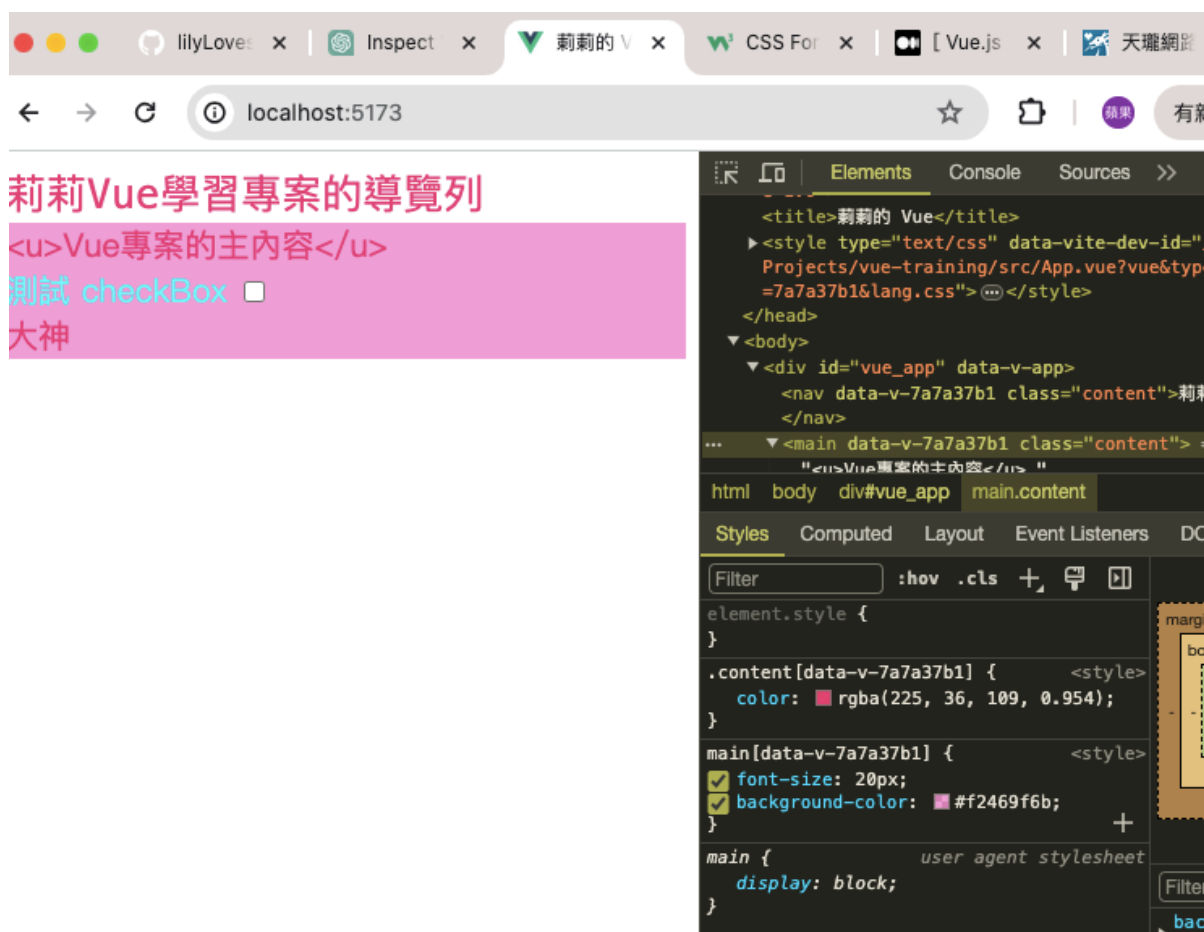
使用 `v-for="(變數, 變數) in 物件"` 語法

組件名稱.vue

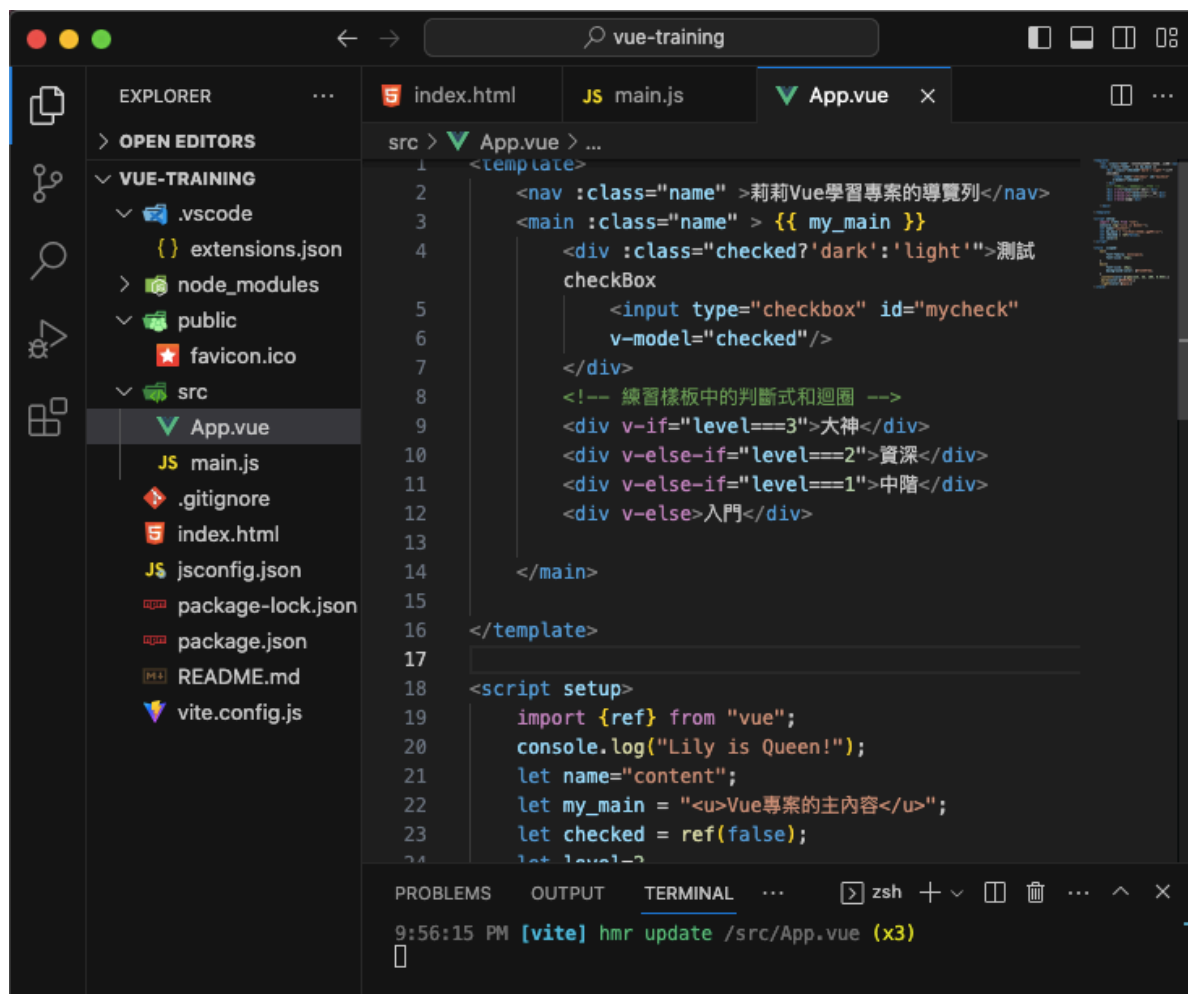
```
<script setup>
  let data={"x":3, "y":4};
</script>
<template>
  <div v-for="(value, key) in data">
    物件屬性 {{ key }} 為 {{ value }}
  </div>
</template>
<style scoped>
  /* CSS 設定 */
</style>
```

#### 7. 開始寫程式: if 判斷式：

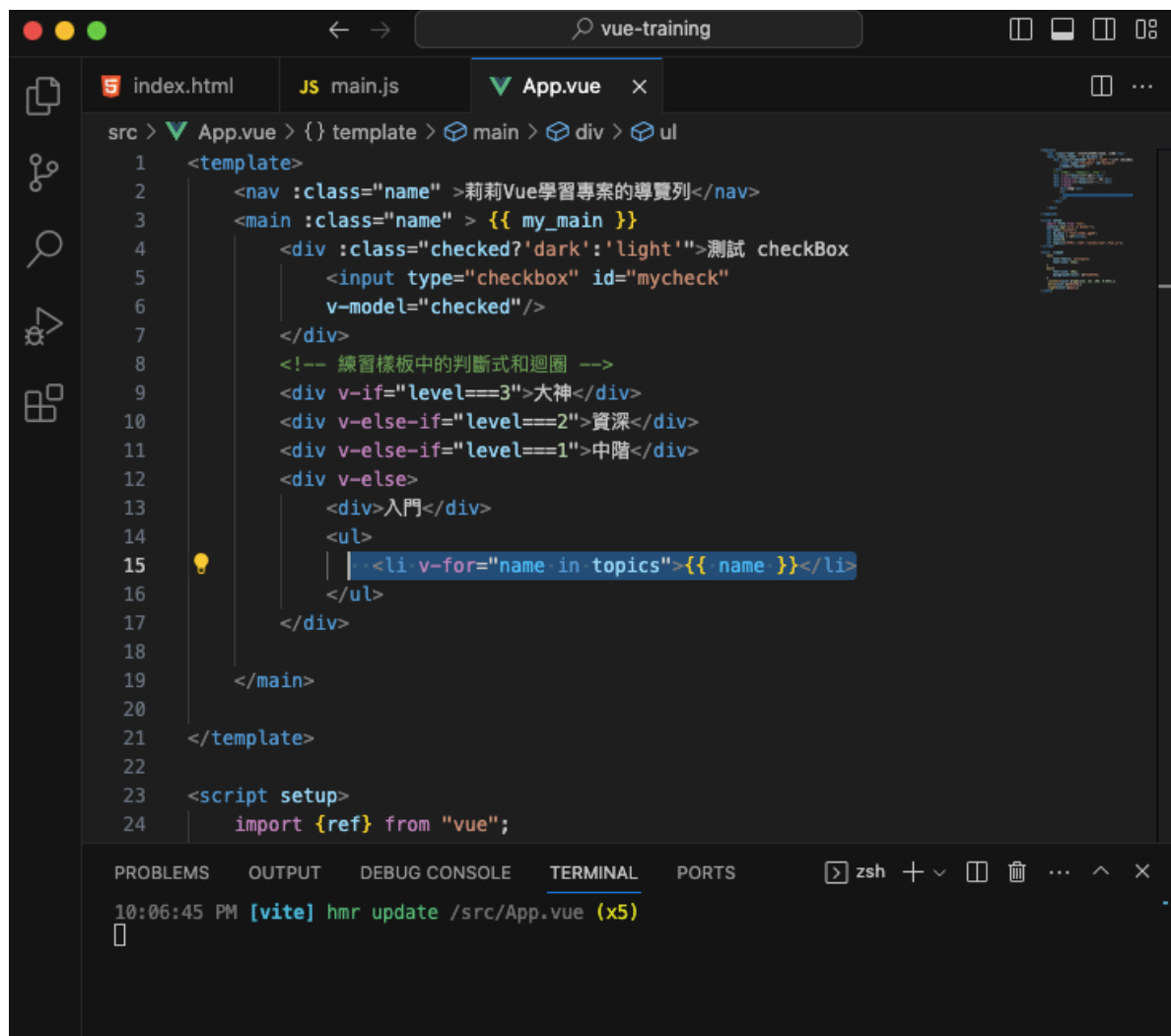
```
src > App.vue > {} template > main > div
1 <template>
2   <nav :class="name" >莉莉Vue學習專案的導覽列</nav>
3   <main :class="name" > {{ my_main }}
4     <div :class="checked?'dark':'light'">測試
      checkBox
5       <input type="checkbox" id="mycheck"
6         v-model="checked"/>
7     </div>
8     <!-- 練習樣板中的判斷式和迴圈 -->
9     <div v-if="level===3">大神</div>
10
11   </main>
12
13 </template>
14
15 <script setup>
16   import {ref} from "vue";
17   console.log("Lily is Queen!");
18   let name="content";
19   let my_main = "<u>Vue專案的主內容</u>";
20   let checked = ref(false);
21   let level=3
22 </script>
23
```

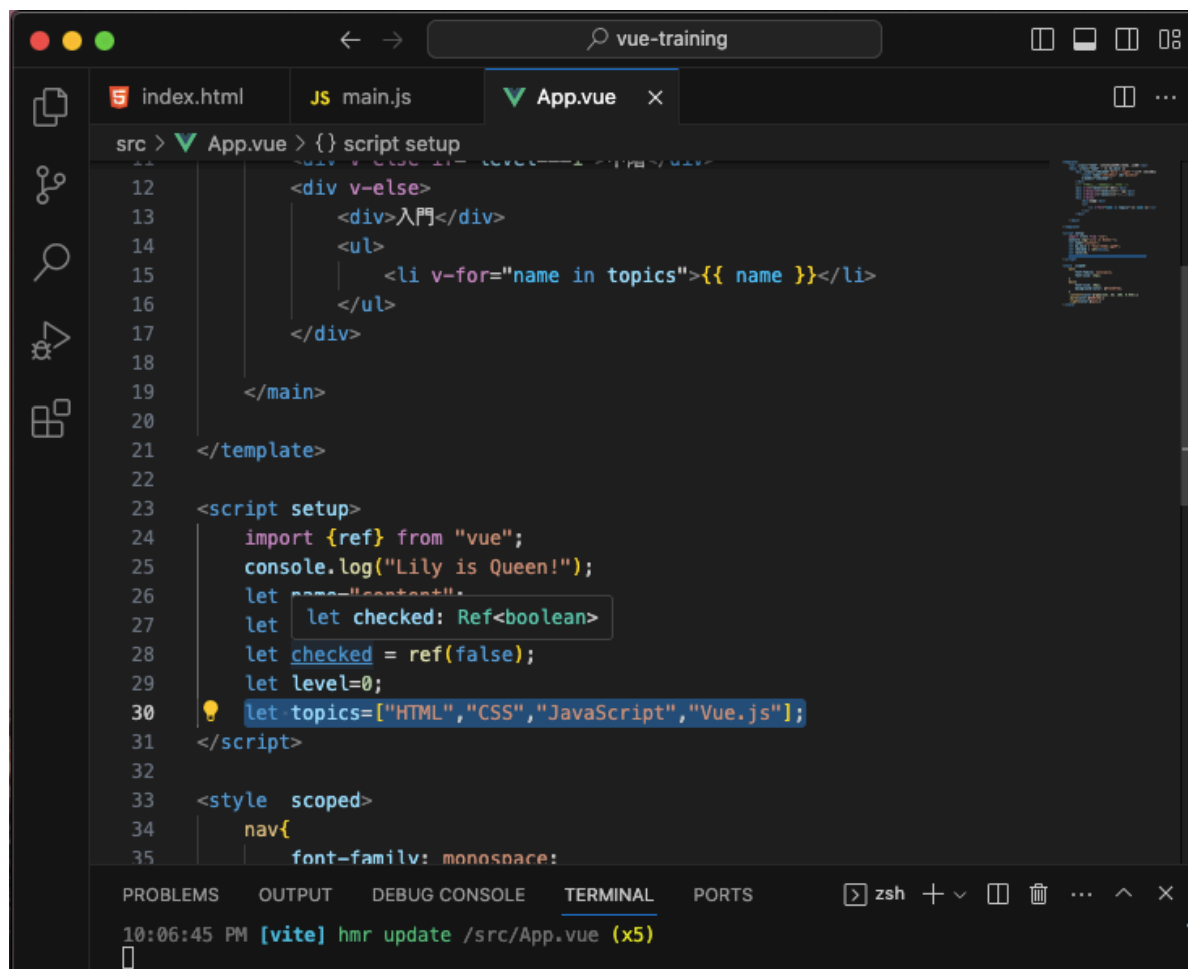


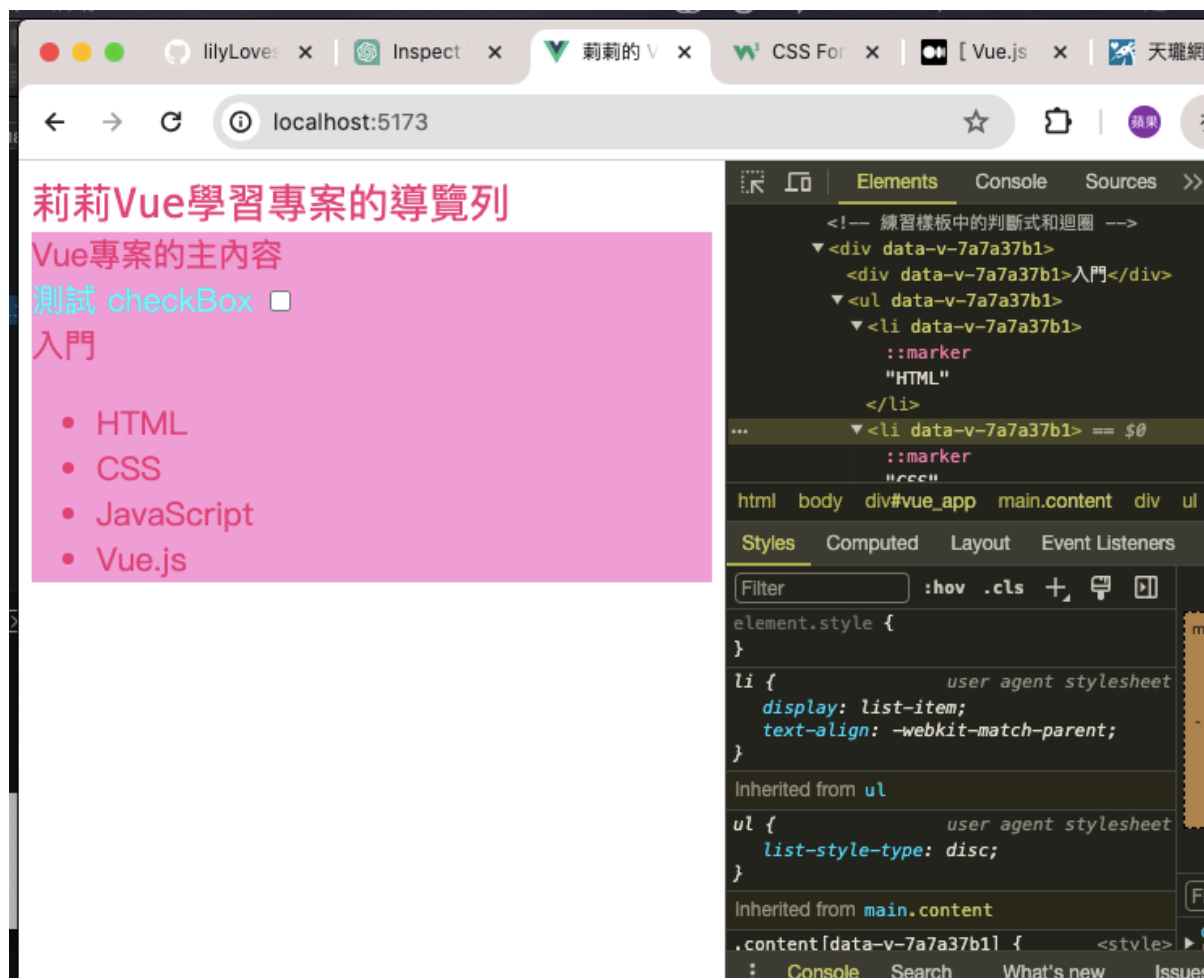
v-if .... v-else-if ... v-else



8. for迴圈 – 陣列 ， 範例:

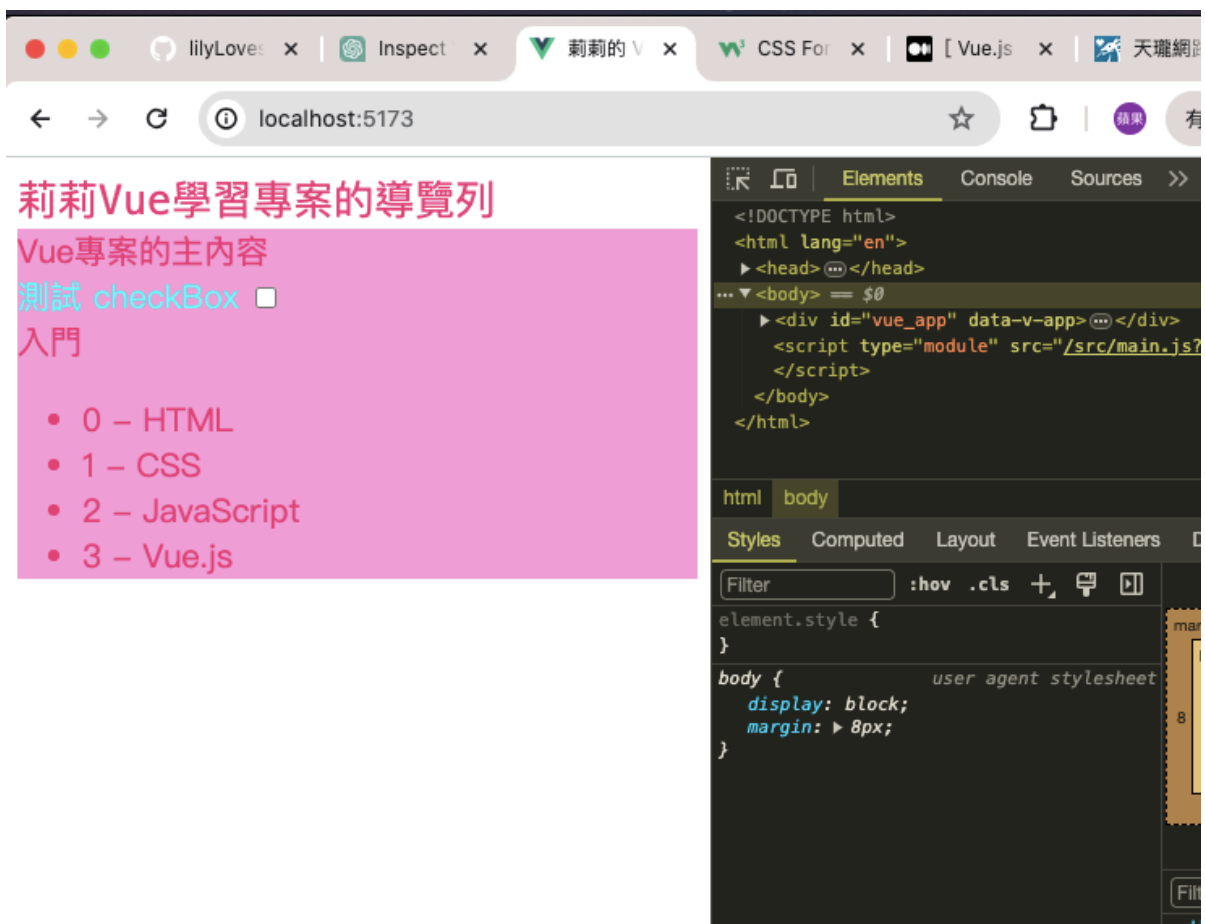






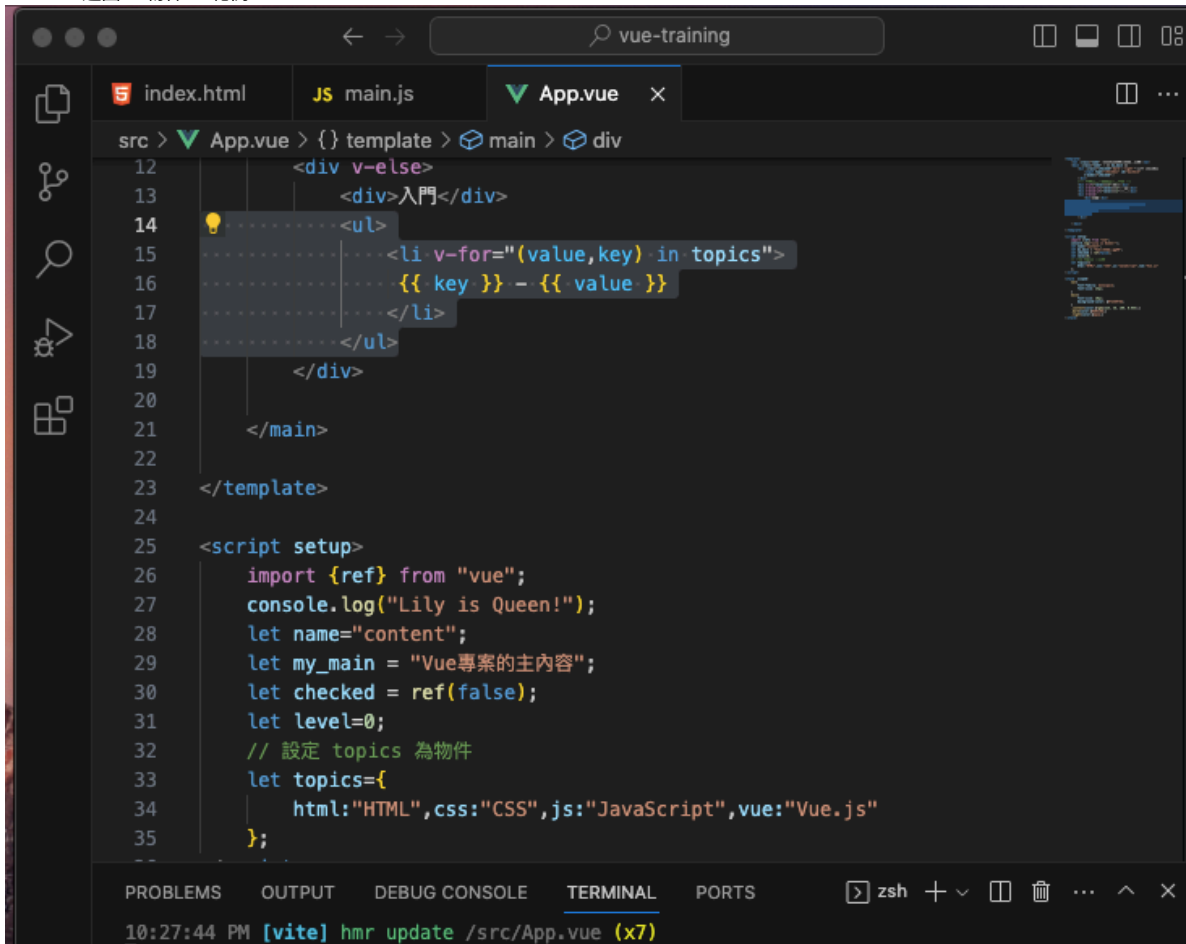
8. for 迴圈 – 陣列索引，範例:

```
vue-training
index.html JS main.js App.vue
src > App.vue > {} template > main > div > ul
10 <div v-else-if="level===2">資深</div>
11 <div v-else-if="level===1">中階</div>
12 <div v-else>
13   <div>入門</div>
14   <ul>
15     <li v-for="(name,index) in topics">
16       {{ index }} - {{ name }}
17     </li>
18   </ul>
19 </div>
20
21 </main>
22
23 </template>
24
25 <script setup>
26   import {ref} from "vue";
27   console.log("Lily is Queen!");
28   let name="content";
29   let my_main = "Vue專案的主內容";
30   let checked = ref(false);
31   let level=0;
32   let topics=["HTML","CSS","JavaScript","Vue.js"];
33 </script>
```





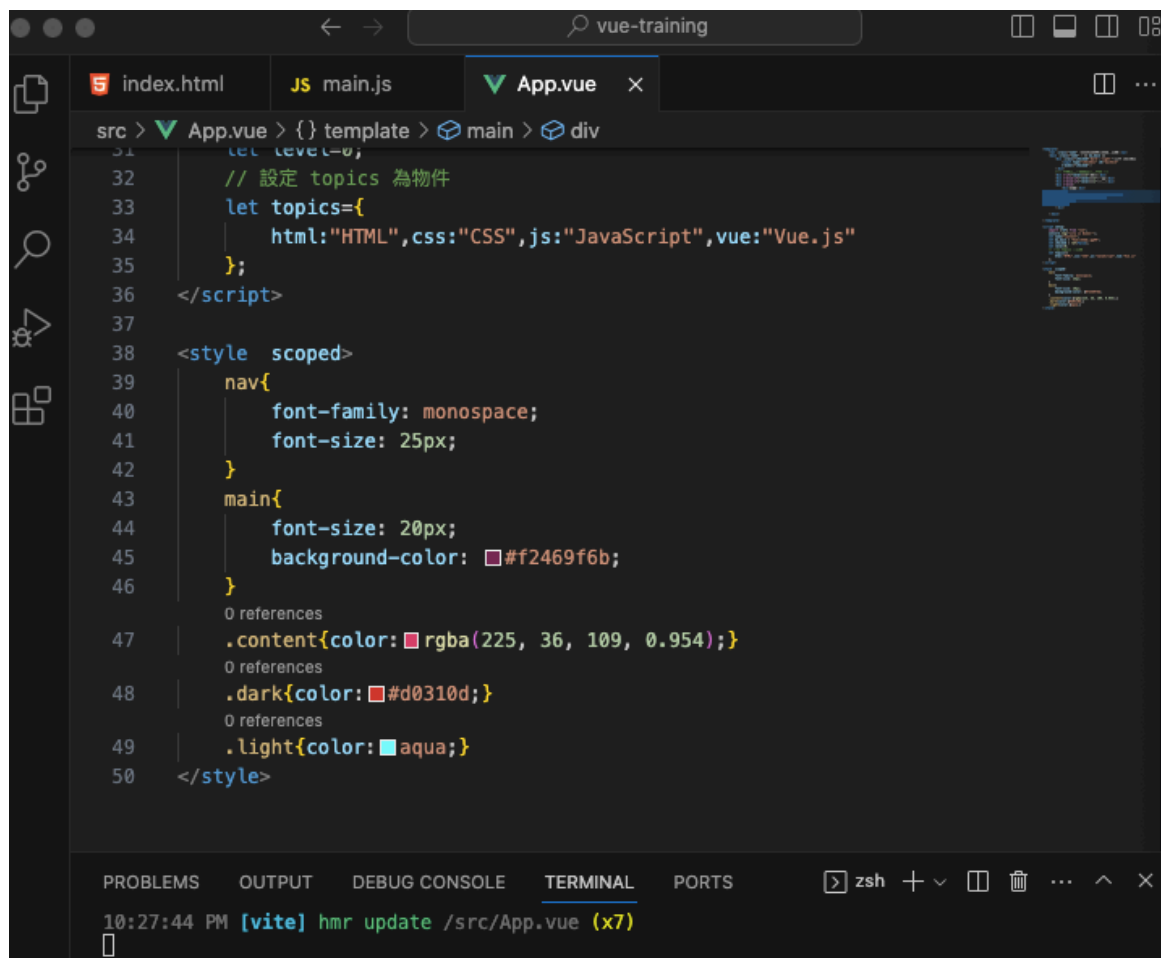
9. 6. for 迴圈 – 物件，範例:



The screenshot shows a code editor with a dark theme. The top bar indicates the project name is 'vue-training'. The file explorer on the left shows a folder 'src' containing 'App.vue', 'main.js', and 'index.html'. The breadcrumb navigation shows the current file is 'App.vue' and the current view is 'template' > 'main' > 'div'. The code is as follows:

```
12     <div v-else>
13       <div>入門</div>
14     <ul>
15       <li v-for="(value,key) in topics">
16         {{ key }} - {{ value }}
17       </li>
18     </ul>
19   </div>
20
21 </main>
22
23 </template>
24
25 <script setup>
26   import {ref} from "vue";
27   console.log("Lily is Queen!");
28   let name="content";
29   let my_main = "Vue專案的主內容";
30   let checked = ref(false);
31   let level=0;
32   // 設定 topics 為物件
33   let topics={
34     html:"HTML",css:"CSS",js:"JavaScript",vue:"Vue.js"
35   };
36 }
```

The bottom of the editor shows a terminal with the command `[vite] hmr update /src/App.vue (x7)` and the time `10:27:44 PM`.





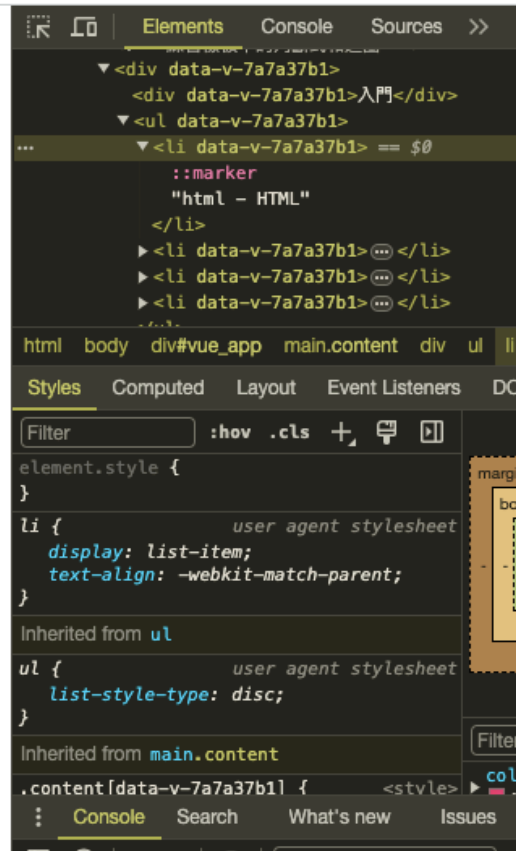
## 莉莉Vue學習專案的導覽列

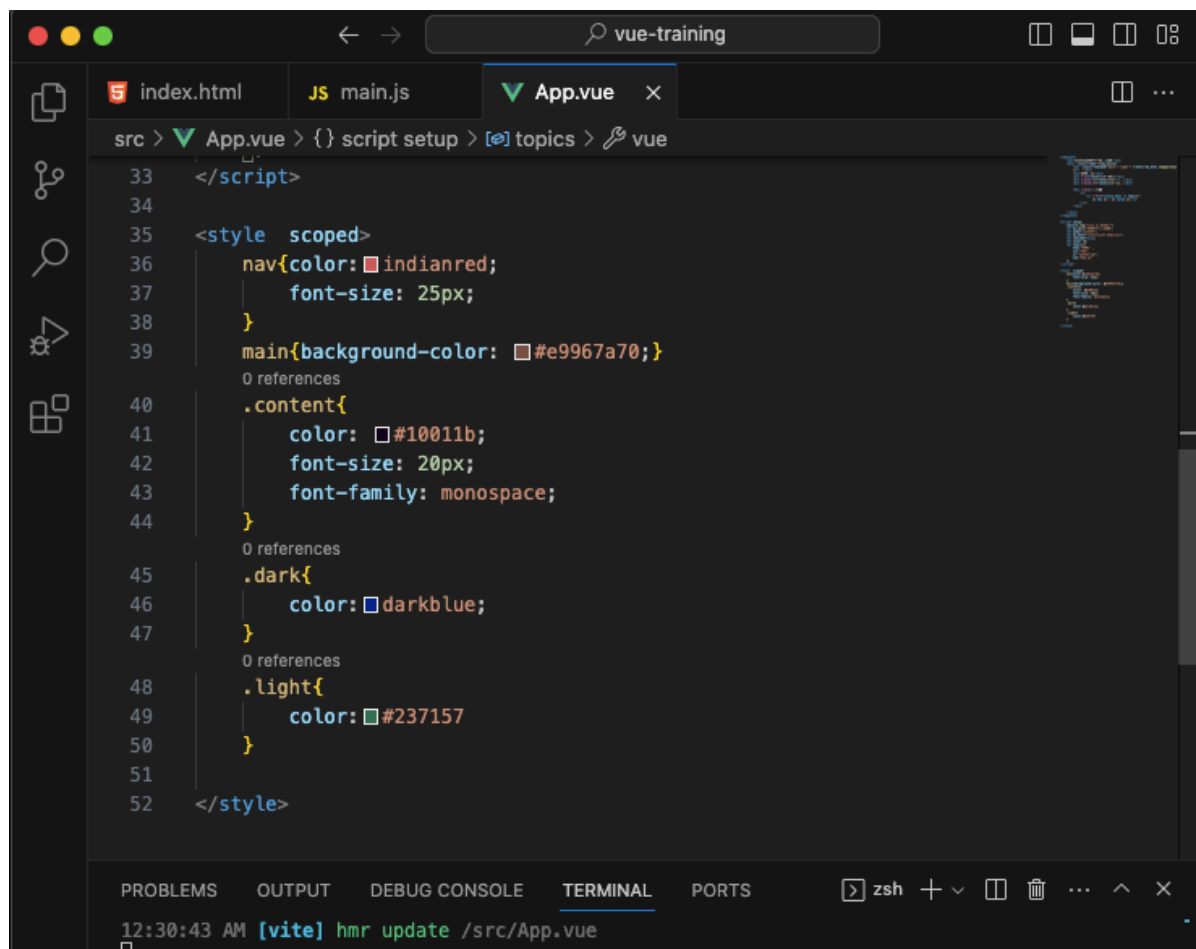
Vue專案的主內容

測試 ☒ checkBox ☐

入門

- html – HTML
- css – CSS
- js – JavaScript
- vue – Vue.js





## 莉莉Vue學習歷程的導覽列

學習歷程主畫面

我在測試 HTML

學習階段

基本功

- html - HTML
- css - CSS
- js - JavaScript
- vue - Vue.js

Elements Console Sources >>

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <div id="vue_app" data-v-app>
      <nav data-v-7a7a37b1>莉莉Vue學習歷程的導覽
    ...
    <main data-v-7a7a37b1 class="content">
      "學習歷程主畫面 "
      <div data-v-7a7a37b1 class="light">...
      <div data-v-7a7a37b1>學習階段</div>
      <div data-v-7a7a37b1>...</div>
    </main>
  </body>
</html>
```

html body div#vue\_app main.content

Styles Computed Layout Event Listeners D

Filter :hov .cls + - [ ]

```
element.style {
}

.content[data-v-7a7a37b1] { <style>
  color: #10011b;
  font-size: 20px;
  font-family: monospace;
}

main[data-v-7a7a37b1] { <style>
  background-color: #e9967a70;
}

main { user agent stylesheet
  display: block;
}
```