

Vue 組件結構：自訂事件處理

1. 組件的切割與載入：

## 組件的切割與載入

App.vue

```
<script setup>
  import Nav from './Nav.vue';
  import Main from './Main.vue';
</script>
<template>
  <Nav></Nav>
  <Main></Main>
</template>
<style scoped>
</style>
```

Nav

```
<script setup>
  console.log("導覽列相")
</script>
<template>
  <nav>
    網頁導覽列
  </nav>
</template>
<style scoped>
  nav{font-weight:bold}
</style>
```

Main

```
<script setup>
  console.log("主區導相")
</script>
<template>
  <main>
    <div>網頁主...
    <div>網頁主...
  </main>
</template>
<style scoped>
  main{padding:30px Op
</style>
```

0:38 / 33:44

2. 組件的使用，標準事件處理

## 組件的使用，標準事件處理

App.vue

```
<script setup>
  import Nav from './Nav.vue';
  import Main from './Main.vue';
  import {ref} from 'vue';
  let text=ref("測試文字");
  let change=function(){
    text.value="修改文字";
  };
</script>
<template>
  <Nav title="任意的標題"></Nav>
  <Main></Main>
  <div>{{ text }}</div>
  <button @click="change">按鈕</button>
</template>
<style scoped>
</style>
</style>
```

暫時忽略 Nav 組件的組件

Ma

```
<script setup>
</script>
<template>
  <main>
    <div>網頁
    <div>網頁
  </main>
</template>
<style scoped>
  main{padding:30px;}
</style>
```

0:56 / 33:44

3. 當 button 的按鈕 擺到 Main.vue 時會發生什麼事？ change 函式就抓不到了，因為 change 定義在 App.vue

## 組件的使用，遇到問題



4. 解決這個問題，使用「自訂事件處理」，在主組件中定義一個事件 @myevent，實際觸發事件是在 main.vue，所以在 Main.vue 寫一個 `$emit('myevent')` → vue 內定的工具，即可解決這個問題

## 組件的使用，自訂事件處理

App.vue

```
<script setup>
  import Nav from './Nav.vue';
  import Main from './Main.vue';
  import {ref} from 'vue';
  let text=ref("測試文字");
  let change=function(){
    text.value="修改文字";
  };
</script>
<template>
  <Nav title="任意的標題"></Nav>
  <Main @myevent="change"></Main>
  <div>{{ text }}</div>
</template>
<style scoped>
</style>
```

Nav

暫時忽略 Nav 組件的細

Main

```
<script setup>
</script>
<template>
  <main>
    <div>網頁
    <div>網頁
    <button @
      按鈕
    </button>
  </main>
</template>
<style scoped>
  main{padding:30
</style>
```

4:30 / 33:44

5. 建立自訂事件的語法: 使用@ 符號加上「自訂事件名稱」

## 建立組件的自訂事件

使用 @ 符號加上自訂事件名稱

- 載入組件  
`import 組件名稱 from`
- 在樣板中使用載入的  
`<組件名稱`  
`@自訂事件名稱`  
`@自訂事件名稱`  
`...`  
`></組件名稱>`
- 基本範例  

```

<script setup>
  import Nav from
  let 函式名稱 =
</script>
<template>
  <Nav @handl
</template>

```

6. 定義好自訂事件，可以用 `$emit`（‘自訂事件名稱’）呼叫，可以「往上父層」去找到自定義事件

組件內部呼叫自訂事件  
方法一

樣板語法的標準事件中使用  
`$emit("自訂事件名稱")`

```
<script setup>
  import Nav from
  let 函式名稱=fn
</script>
<template>
  <Nav @handler
</template>
```

```
<script setup>
</script>
<template>
  <nav>
    <button
      按
    </button>
  </nav>
</template>
```

9:34 / 33:44

7. 在組件內部呼叫 自訂事件 方法二:

組件內部呼叫自訂事件  
方法二

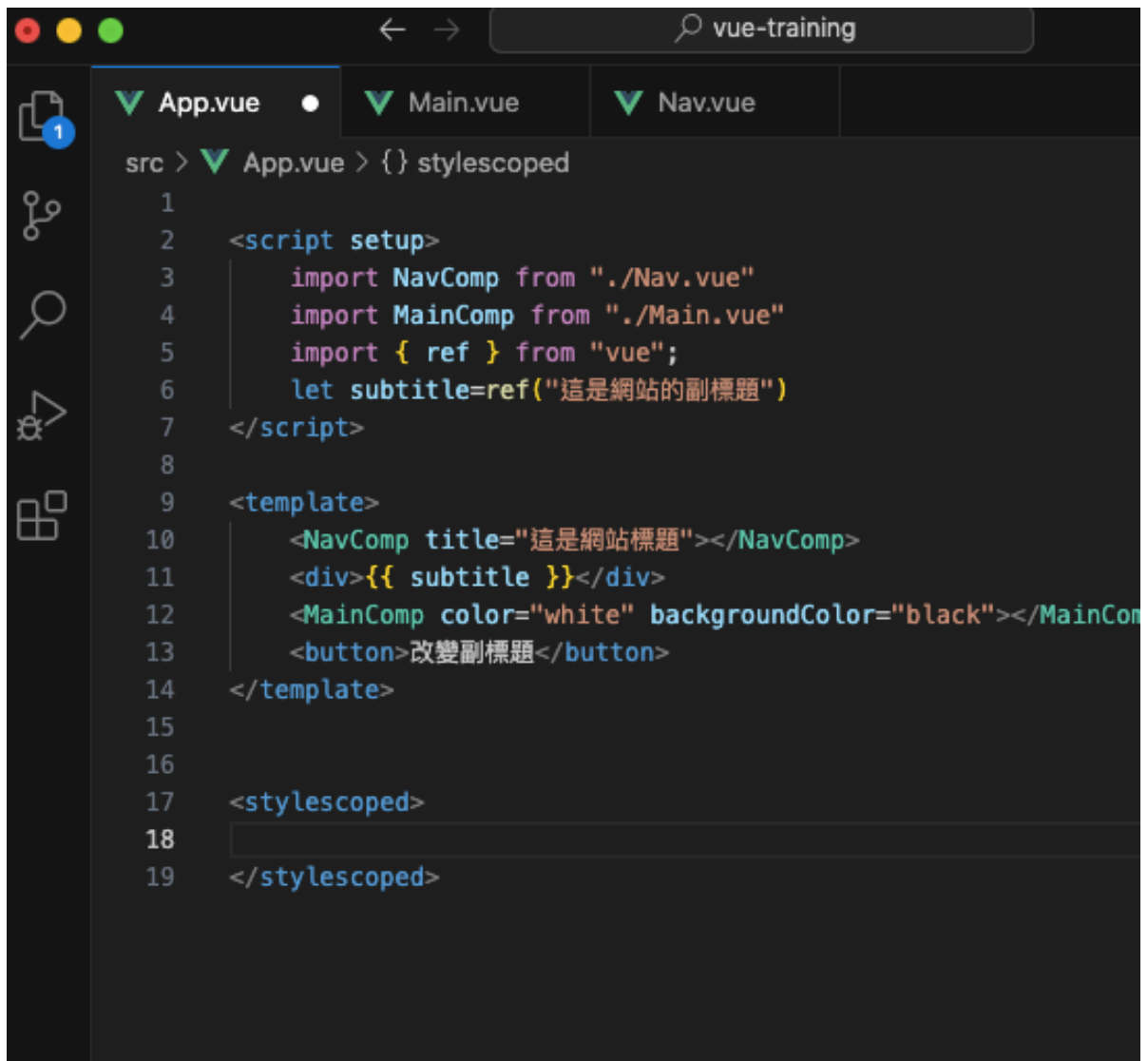
JavaScript 程式中使用  
`defineEmits(["自訂事件名稱", ...])`

11:23 / 33:44

```
<script setup>
  import Nav from "/N
  let 函式名稱=function
</script>
<template>
  <Nav @handler="函:
</template>
```

```
<script setup>
  let emit=defineEmits
  let 函式名稱=function
</script>
<template>
  <nav>
    <button @click
  </nav>
</template>
```

8. 寫點程式：在標題底下，加一個副標題，然後加一個按鈕 改變副標題



```
src > App.vue > {} scoped
1
2  <script setup>
3    import NavComp from "../Nav.vue"
4    import MainComp from "../Main.vue"
5    import { ref } from "vue";
6    let subtitle=ref("這是網站的副標題")
7  </script>
8
9  <template>
10    <NavComp title="這是網站標題"></NavComp>
11    <div>{{ subtitle }}</div>
12    <MainComp color="white" backgroundColor="black"></MainComp>
13    <button>改變副標題</button>
14  </template>
15
16
17  <scoped>
18
19  </scoped>
```



← → vue-training

App.vue

Main.vue

Nav.vue

src > App.vue > ...

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

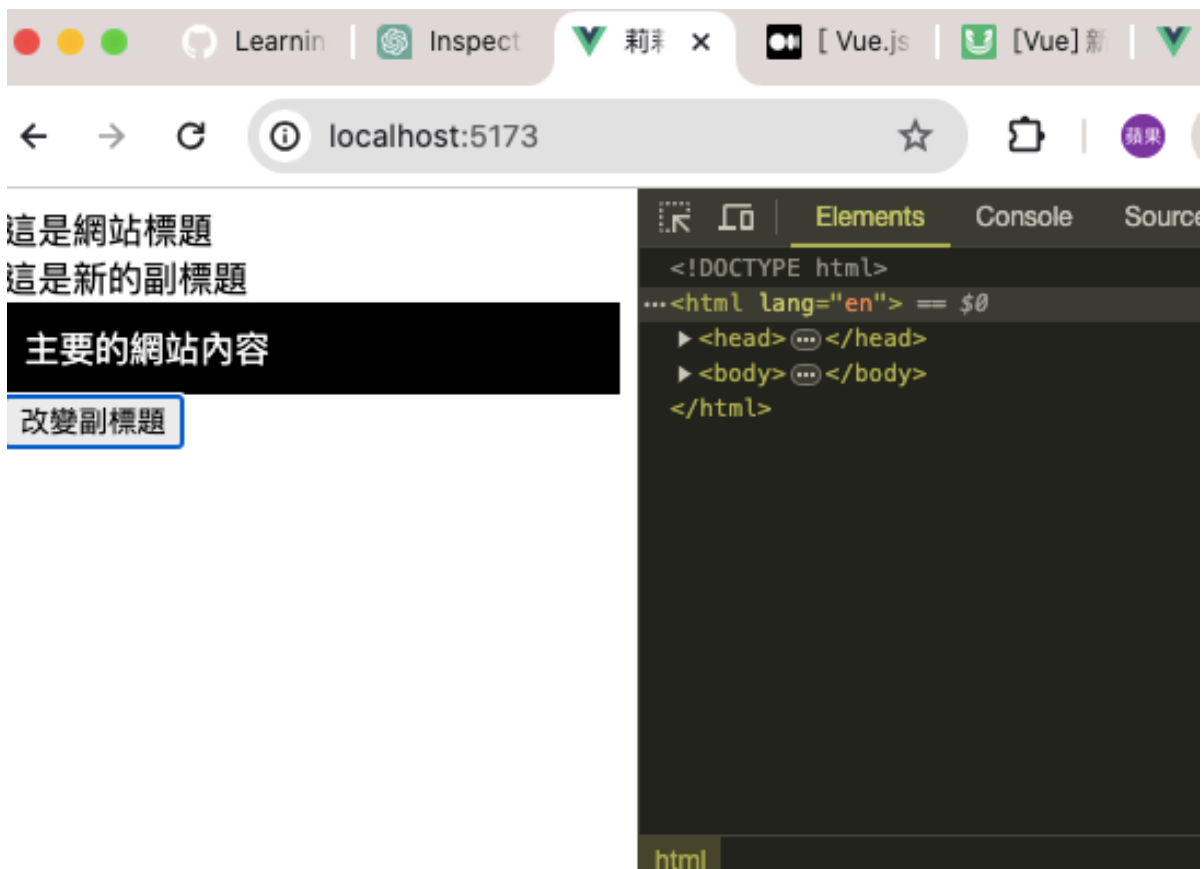
```
<script setup>
  import NavComp from "./Nav.vue"
  import MainComp from "./Main.vue"
  import { ref } from "vue";
  let subtitle=ref("這是網站的副標題")
  let changeText =(changeSubtitle)=>{
    subtitle.value=changeSubtitle;
  };
</script>

<template>
  <NavComp title="這是網站標題"></NavComp>
  <div>{{ subtitle }}</div>
  <MainComp color="white" backgroundColor="black"></MainComp>
  <button @click = "changeText('這是新的副標題')">改變副標題</button>
</template>

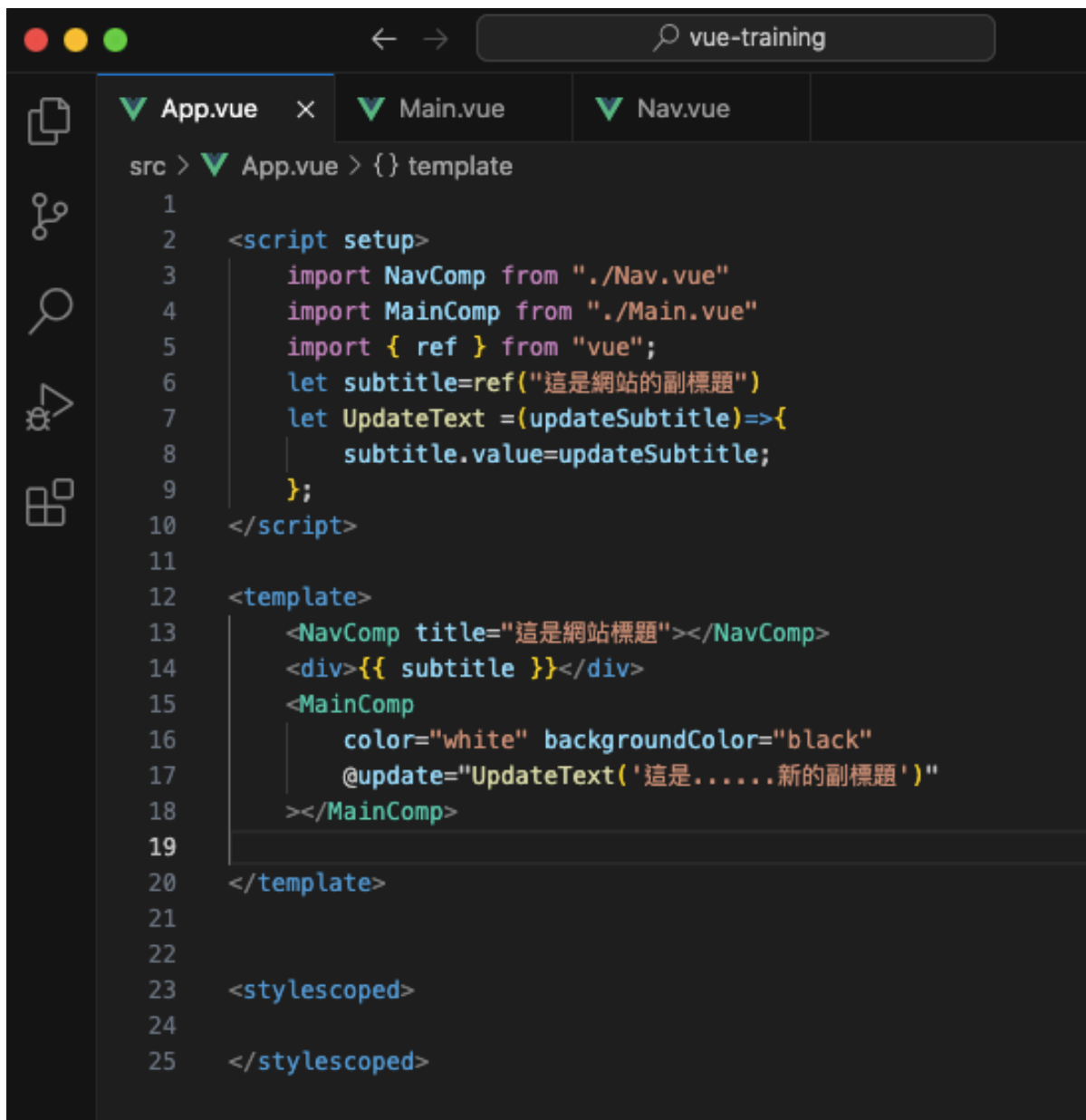
<style>
</style>
</style>
```



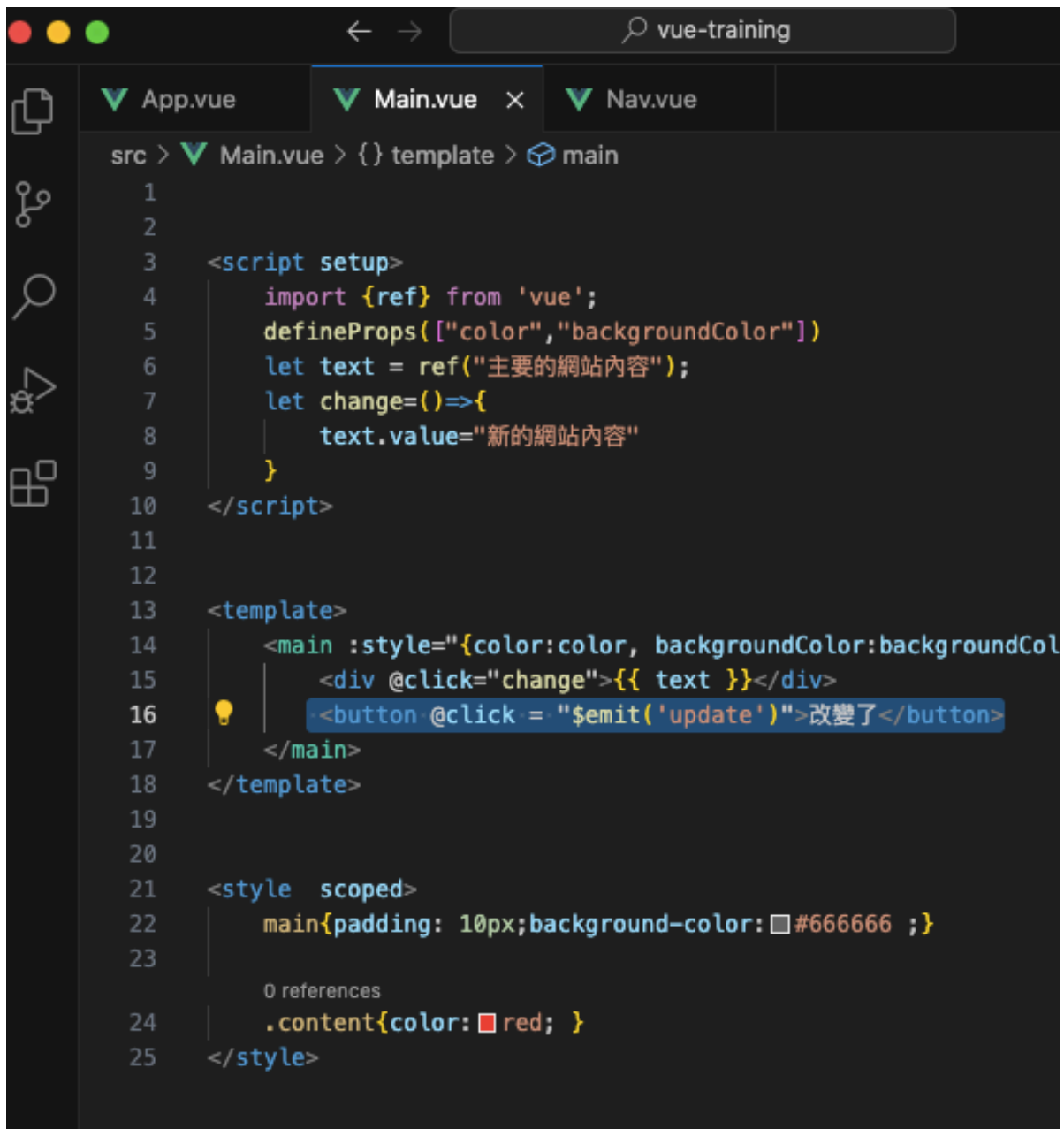
當按下「改變副標題」：



9. 現在有一個問題，就是 要把「按鈕」放到 Main.vue 該怎麼辦？放過去之後，按鈕就無效了，於是要 用到自訂義事件：在 App.vue 用 @ 自訂義一個事件，然後在 Main.vue 用 \$emit 呼叫



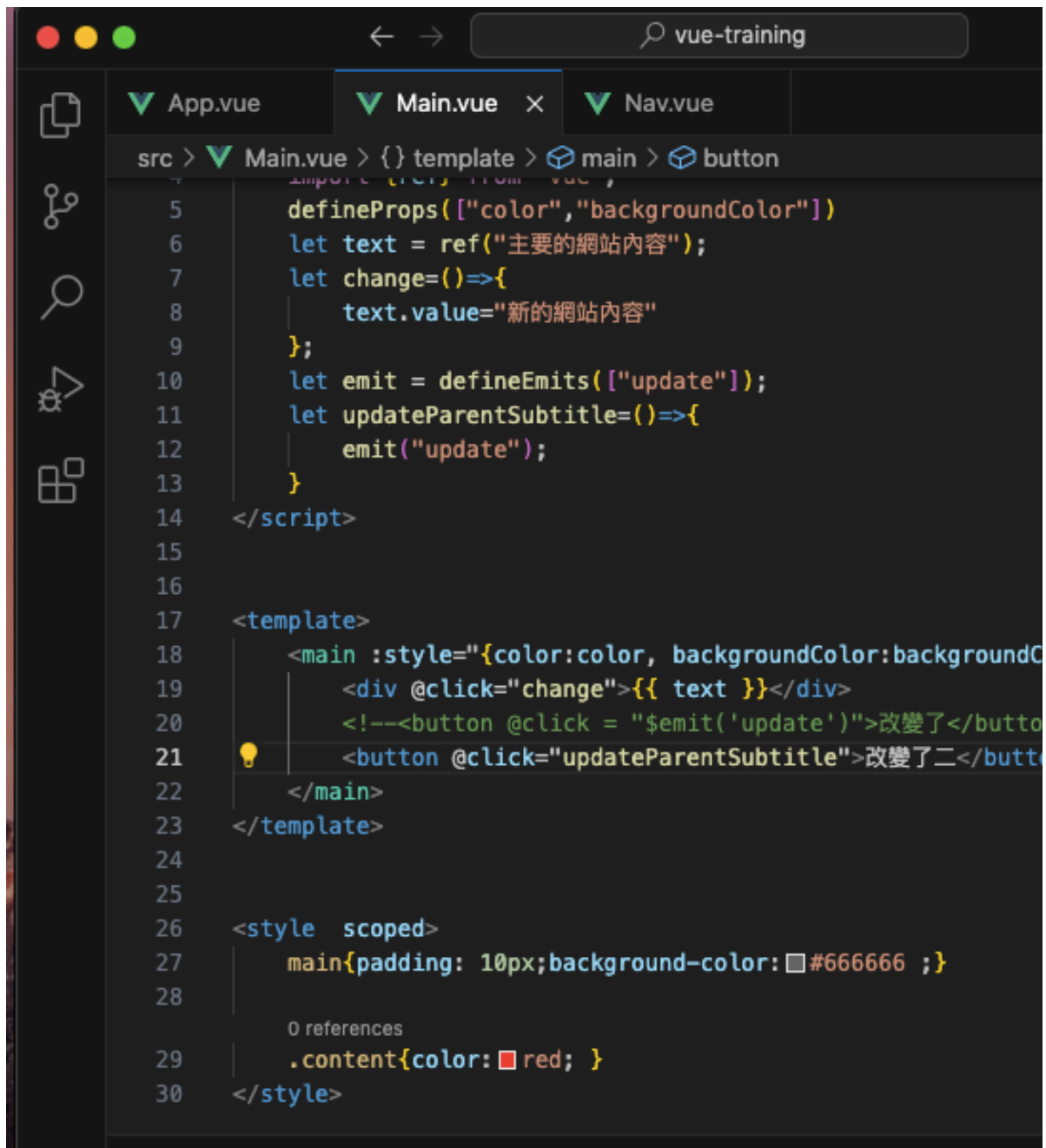
```
src > App.vue > {} template
1
2 <script setup>
3   import NavComp from "../Nav.vue"
4   import MainComp from "../Main.vue"
5   import { ref } from "vue";
6   let subtitle=ref("這是網站的副標題")
7   let UpdateText =(updateSubtitle)=>{
8     subtitle.value=updateSubtitle;
9   };
10 </script>
11
12 <template>
13   <NavComp title="這是網站標題"></NavComp>
14   <div>{{ subtitle }}</div>
15   <MainComp
16     color="white" backgroundColor="black"
17     @update="UpdateText( '這是.....新的副標題' )"
18   ></MainComp>
19
20 </template>
21
22
23 <style scoped>
24
25 </style>
```



```
src > App.vue Main.vue x Nav.vue
1
2
3 <script setup>
4   import {ref} from 'vue';
5   defineProps(["color","backgroundColor"])
6   let text = ref("主要的網站內容");
7   let change=()=>{
8     text.value="新的網站內容"
9   }
10 </script>
11
12
13 <template>
14   <main :style="{color:color, backgroundColor:backgroundCol
15     <div @click="change">{{ text }}</div>
16     <button @click="$emit('update')">改變了</button>
17   </main>
18 </template>
19
20
21 <style scoped>
22   main{padding: 10px;background-color: #666666 ;}
23
24   0 references
25   .content{color: red; }
26 </style>
```

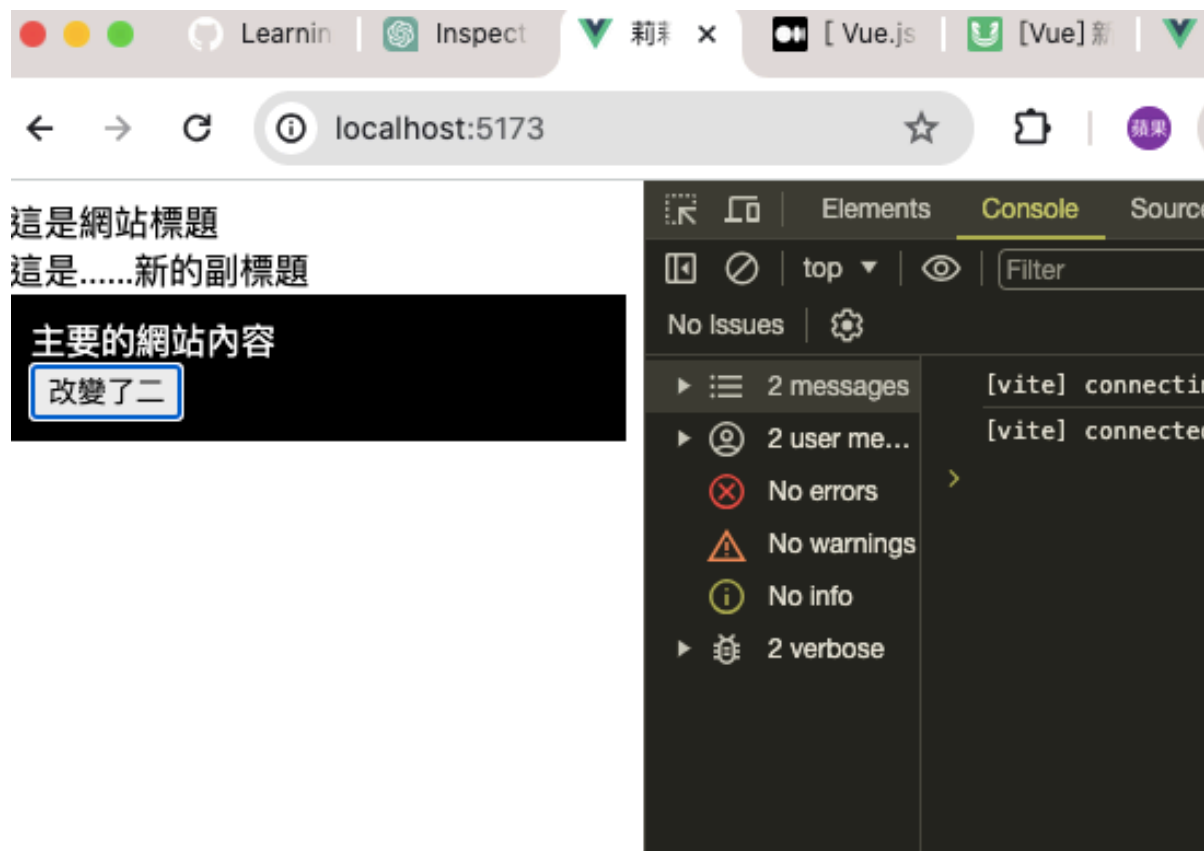
10.自訂義事件 方法二：在Main.vue 的 script 裏用 Vue 的函式 自訂義事件(用陣列概念) 回傳一個 emit 方法:

```
let emit = defineEmits(["update"]);
```



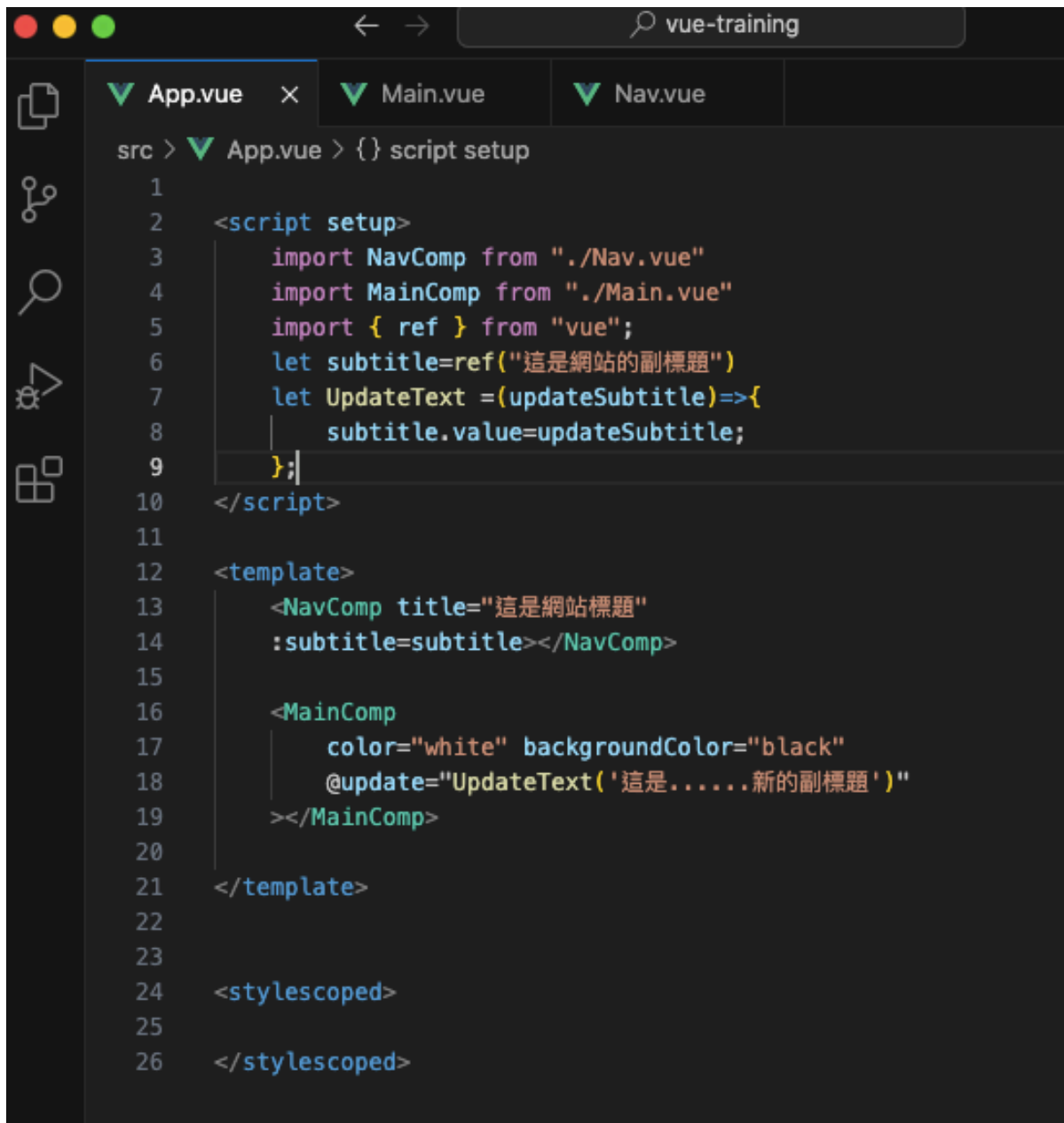
```
src > ▼ Main.vue > {} template > main > button
5     defineProps(["color","backgroundColor"])
6     let text = ref("主要的網站內容");
7     let change=()=>{
8       text.value="新的網站內容"
9     };
10    let emit = defineEmits(["update"]);
11    let updateParentSubtitle=()=>{
12      emit("update");
13    }
14  </script>
15
16
17  <template>
18    <main :style="{color:color, backgroundColor:backgroundC
19      <div @click="change">{{ text }}</div>
20      <!--<button @click = "$emit('update')">改變了</butto
21    <button @click="updateParentSubtitle">改變了二</butt
22    </main>
23  </template>
24
25
26  <style scoped>
27    main{padding: 10px;background-color: #666666 ;}
28
29    0 references
30    .content{color: red; }
```

方法二，有成功：

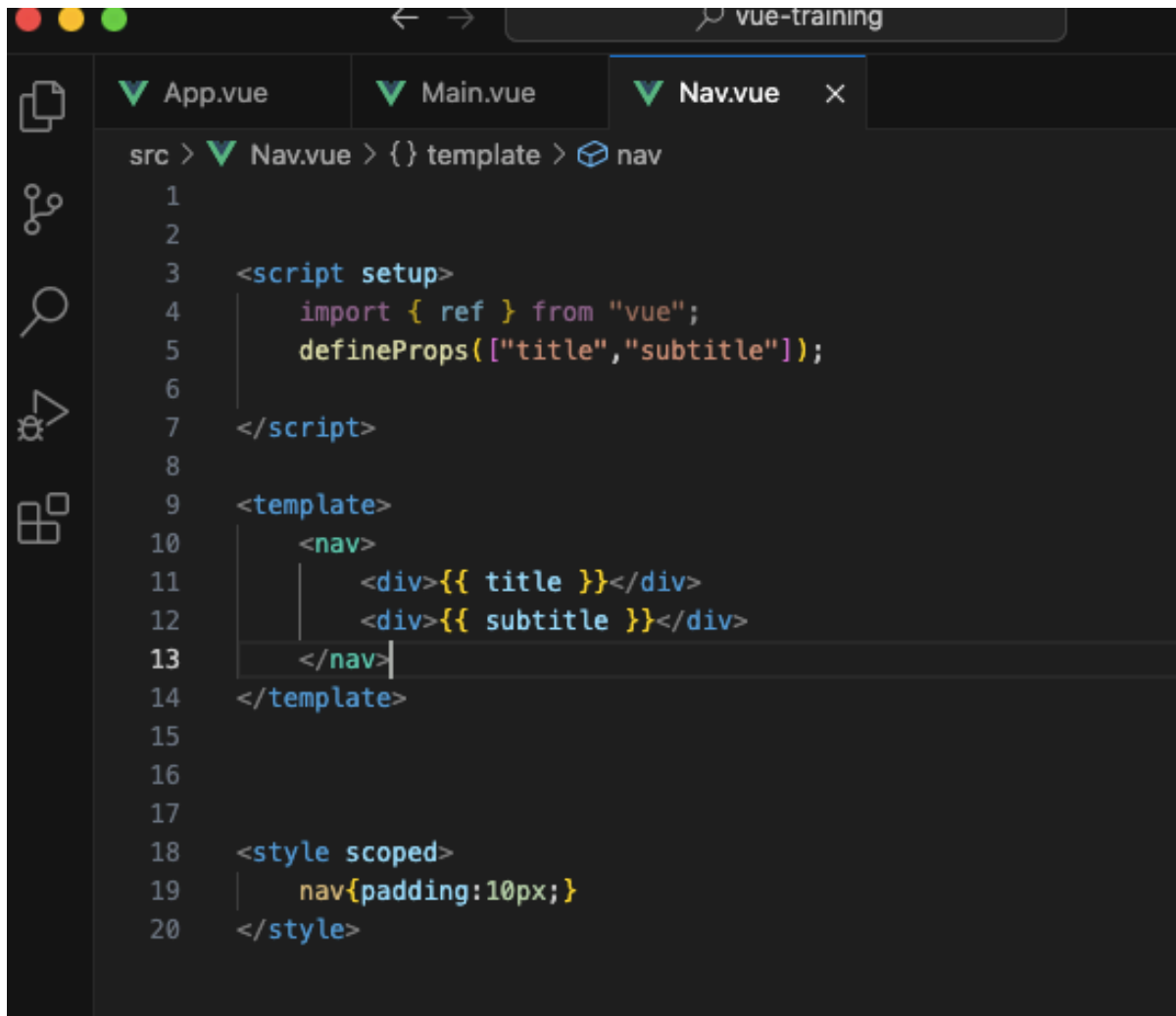


11. 打 subtitle 加到 Nav.vue :

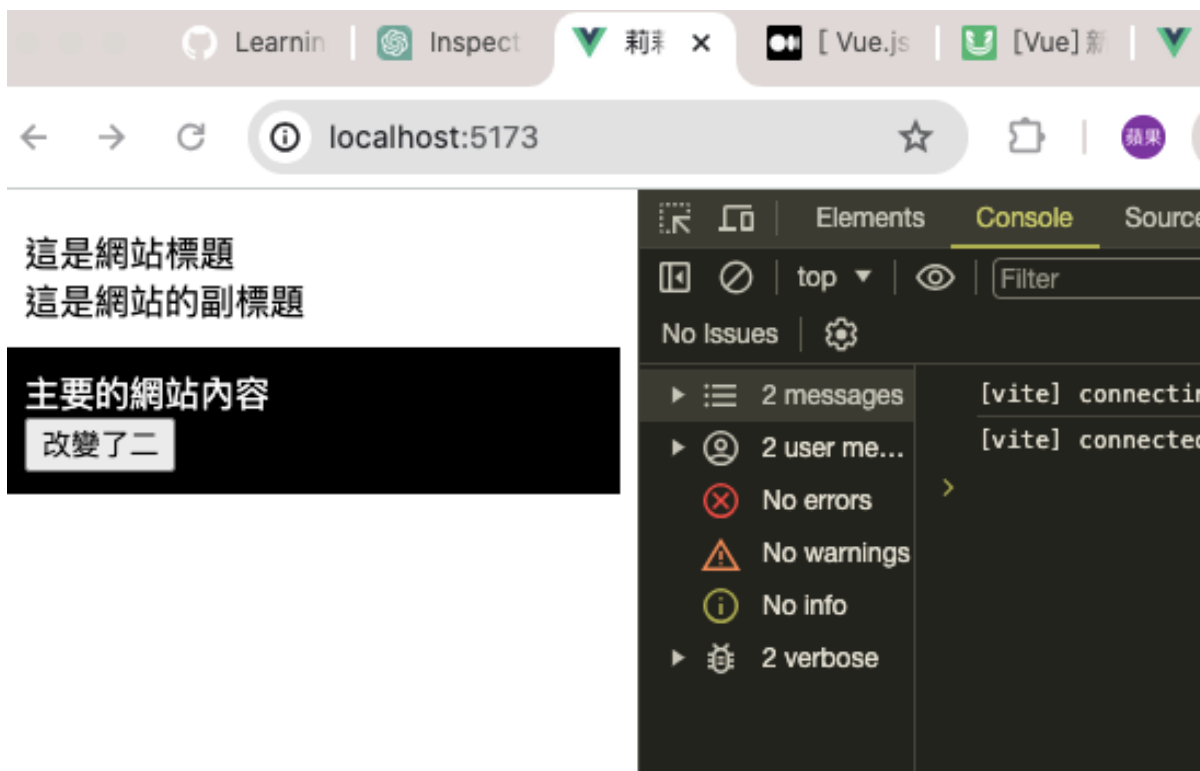
在 template 的 subtitle ,前面要加 `:` , 才會連動  
:



```
src > App.vue > {} script setup
1
2 <script setup>
3   import NavComp from "../Nav.vue"
4   import MainComp from "../Main.vue"
5   import { ref } from "vue";
6   let subtitle=ref("這是網站的副標題")
7   let UpdateText =(updateSubtitle)=>{
8     subtitle.value=updateSubtitle;
9   };
10 </script>
11
12 <template>
13   <NavComp title="這是網站標題"
14     :subtitle=subtitle></NavComp>
15
16   <MainComp
17     color="white" backgroundColor="black"
18     @update="UpdateText('這是.....新的副標題')"
19   ></MainComp>
20
21 </template>
22
23
24 <style scoped>
25
26 </style>
```



```
src > Nav.vue > {} template > nav
1
2
3 <script setup>
4   import { ref } from "vue";
5   defineProps(["title","subtitle"]);
6
7 </script>
8
9 <template>
10   <nav>
11     <div>{{ title }}</div>
12     <div>{{ subtitle }}</div>
13   </nav>
14 </template>
15
16
17
18 <style scoped>
19   nav{padding:10px;}
20 </style>
```





按一下按鈕：

