



40

元 件 能

# Main.vue

```
<template>
  <!-- Define Html Template -->
  <Calendar/>
  <Calendar/>
</template>

<script>
  // Define js logic
  import Calendar from 'Calendar.vue'

  export default {
    name: "Main",
    components:{ Calendar },
    ...
  }
</script>

<style lang="scss" scoped>
  // Define Scss
  ...
</style>
```

September 2017						
MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

September 2017						
MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

2個日曆各自有自己的狀態  
不會被另一個日曆影響

↓ 是與另一個日曆無關

o SuedoCode 為列

# 管理複雜化之前端



- ◎ 以SPA為例
  - 何時更新Data，同步Data？
  - 何時該連動更新畫面？
  - 何時該呼叫後端更新Data？



# 元件化

## Main.vue

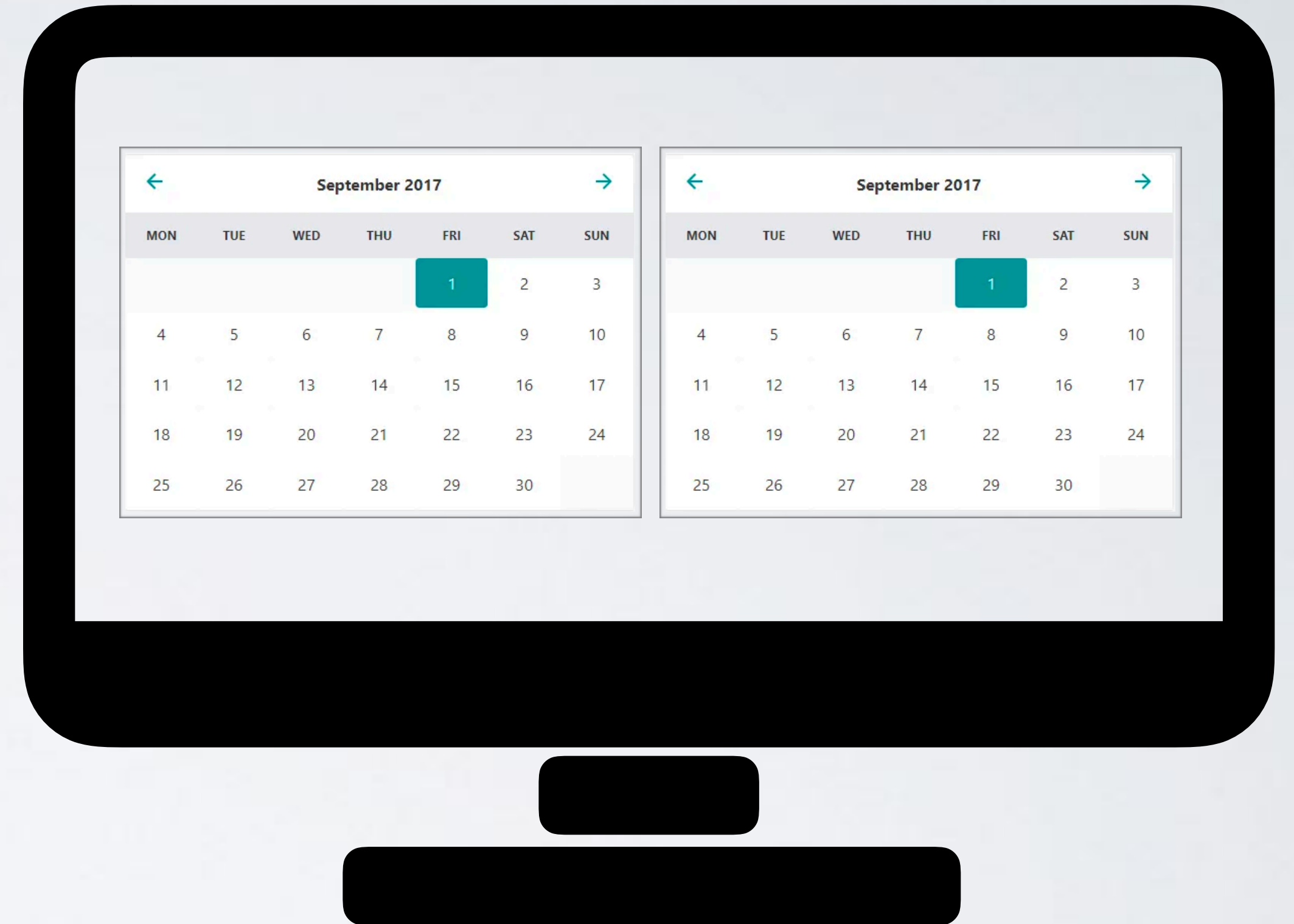
```
<template>
  <!-- Define Html Template -->
  <Calendar/>
  <Calendar/>
</template>

<script>
  // Define js logic
  import Calendar from 'Calendar.vue'

  export default {
    name: "Main",
    components:{ Calendar },
    ...
  }
</script>

<style lang="scss" scoped>
  // Define Scss
  ...
</style>
```

◎ Suedo Code以Vue為例



2個日曆各自有自己的狀態  
不會被另一個日曆影響