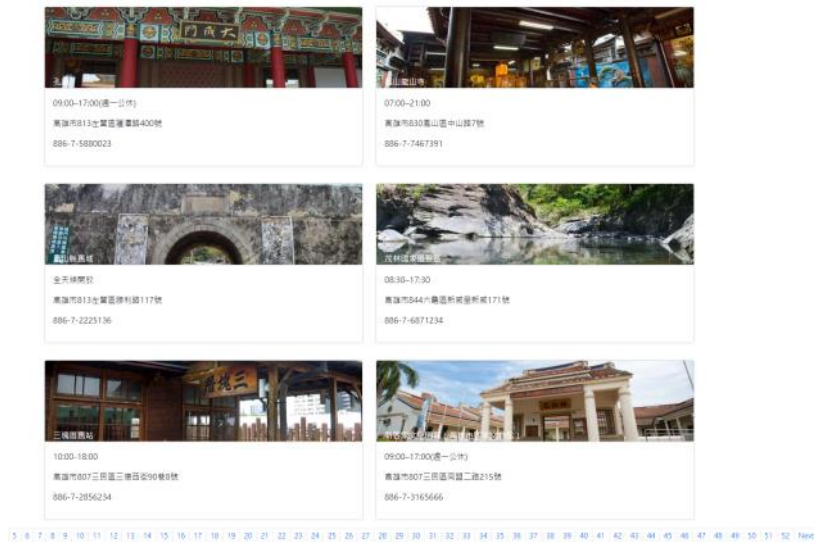


14. Component範例練習

以高雄旅遊網資料製作如下圖排排版，使用components搭配 props 與 emit傳遞參數，點擊page以改變上列景點資料!

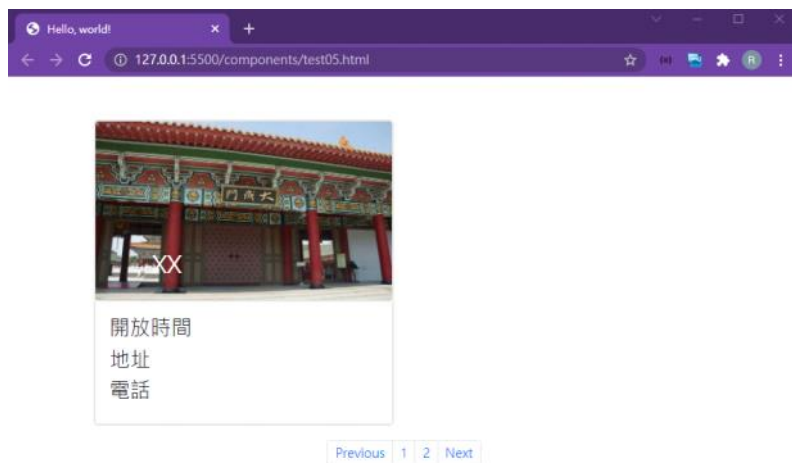


1. 版面製作

以兩層card製作

```
<div class="container" id="app">
  <div class="row">
    <div class="col-md-6">
      <div class="card h-100">
        <div class="card">
          
          <div class="card-img-overlay d-flex align-items-end fw-bold text-white">
            <h3 class="mx-5">XX</h3>
          </div>
        </div>
      </div>
      <div class="card h-100">
        <div class="card-body">
          <h4>開放時間</h4>
          <h4>地址</h4>
          <h4>電話</h4>
        </div>
      </div>
    </div>
  </div>
  <div class="d-flex justify-content-center mt-3">...
```

```
<div class="container" id="app">
  <div class="row py-5">
    <div class="col-md-6">...
  </div>
  <div class="d-flex justify-content-center mt-3">
    <nav aria-label="Page navigation example">
      <ul class="pagination pagination-sm">
        <li class="page-item">
          <a class="page-link" href="#" aria-label="Previous">
            <span aria-hidden="true">Previous</span>
          </a>
        </li>
        <li class="page-item"><a class="page-link" href="#">1</a></li>
        <li class="page-item"><a class="page-link" href="#">2</a></li>
        <li class="page-item">
          <a class="page-link" href="#" aria-label="Next">
            <span aria-hidden="true">Next</span>
          </a>
        </li>
      </ul>
    </nav>
  </div>
</div>
```



2. 載入資料

<https://api.kcg.gov.tw/api/service/Get/9c8e1450-e833-499c-8320-29b36b7ace5c>

定義data

```
<script>
  const App = Vue.createApp({
    data() {
      return {
        jsonUrl: "9c8e1450-e833-499c-8320-29b36b7ace5c.json", //高雄旅遊網api資料
        jsonData: [], //存放高雄旅遊網api資料
        currentPage: 1, //目前的頁面
        perPage: 20, //每頁數量
      }
    },
    methods: { ...
    mounted() { ...
  });
  App.mount('#app');
</script>
```

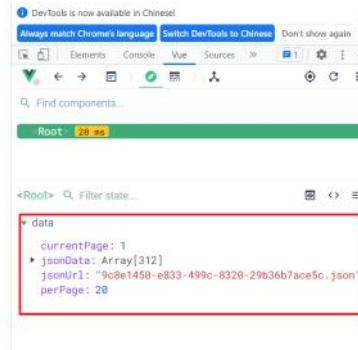
取得外部資料

```
<script>
  const App = Vue.createApp({
    data() { ...
    methods: {
      //取得高雄旅遊網api資料
      getData() {
        // Make a request for a user with a given ID
        const vm = this;
        axios.get(this.jsonUrl)
          .then(function (response) {
            // handle success
            console.log(response.data.data.XML_Head.Infos.Info);
            vm.jsonData = response.data.data.XML_Head.Infos.Info;
          })
          .catch(function (error) {
            // handle error
            console.log(error);
          })
          .then(function () {
            // always executed
          });
      },
    },
    mounted() {
      this.getData();
    },
  });
  App.mount('#app');
</script>
```





Previous 1 2 Next



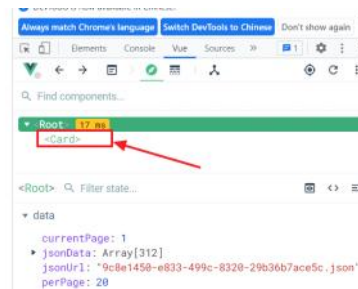
3. 使用components產生card

```
<div class="container" id="app">
  <div class="row pt-5">
    <card></card>
    <div class="d-flex justify-content-center mt-3">...
  </div>
</div>
```

```
App.component('card', {
  template: `<div class="col-md-6">
    <div class="card h-100">
      <div class="card">
        
        <div class="card-img-overlay d-flex align-items-end fw-bold text-white">
          <h3 class="mx-5">XX</h3>
        </div>
      </div>
      <div class="card h-100">
        <div class="card-body">
          <h4>開放時間</h4>
          <h4>地址</h4>
          <h4>電話</h4>
        </div>
      </div>
    </div>
  </div>`
});
```



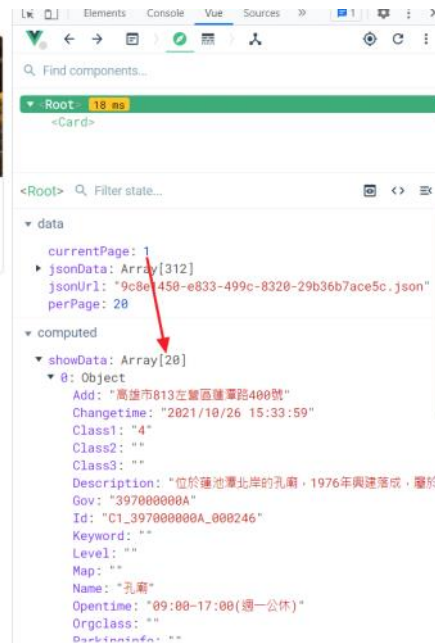
Previous 1 2 Next



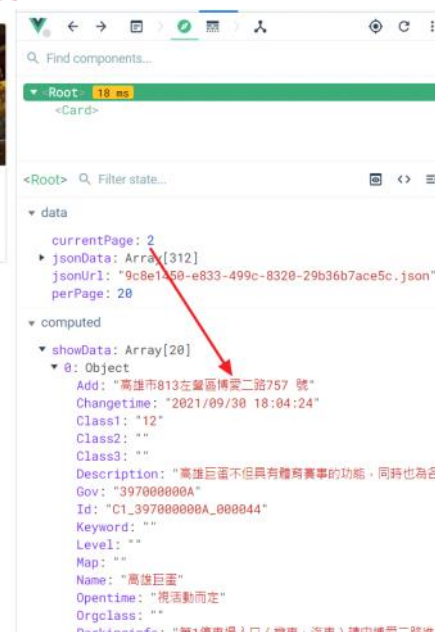
4. 使用 computed 及時挑選每一頁的顯示資料

```
computed: {
  showData() {
    const vm = this;
    const data = [];
    const minData = (vm.currentPage * vm.perPage) - vm.perPage; // 每一頁的第一筆資料
    const maxData = (vm.currentPage * vm.perPage) - 1; // 每一頁的最後一筆資料

    vm.jsonData.forEach((item, key) => {
      if(key >= minData && key <= maxData){
        data.push(item);
      }
    });
    return data;
  }
}
```



currentPage改變觸發showData改變



5. 以props 傳遞資料至 componets

:item="item"

等號左邊的item為要傳入props的參數

等號右邊的item為 v-for所產生的資料

```
<div class="container" id="app">
  <div class="row pt-5">
    <card v-for="(item, key) in showData" :key="item.Id" :item="item"></card>
    <div class="d-flex justify-content-center mt-3">...
  </div>
</div>
```

```

App.component('card', {
  props: ['item'],
  template: `<div class="col-md-6 my-3">
    <div class="card h-100">
      <div class="card">
        
        <div class="card-img-overlay d-flex align-items-end fw-bold text-white">
          <h3 class="mx-5">{{item.Name}}</h3>
        </div>
      </div>
      <div class="card h-100">
        <div class="card-body">
          <h4>{{item.Opentime}}</h4>
          <h4>{{item.Add}}</h4>
          <h4>{{item.Tel}}</h4>
        </div>
      </div>
    </div>
  </div>`
});

```



09:00-17:00(週一公休)
高雄市813左營區蓮潭路400號
886-7-5880023



07:00-21:00
高雄市830鳳山區中山路7號
886-7-7467391



全天候開放
高雄市813左營區勝利路117號
886-7-2225136



08:30-17:30
高雄市844六龜區新威里新威171號
886-7-6871234



10:00-18:00
高雄市807三民區三德西街90巷8號
886-7-2856234



09:00-17:00(週一公休)
高雄市807三民區同盟二路215號
886-7-3165666

[Previous](#) [1](#) [2](#) [Next](#)

6. 使用components產生pagination

```

App.component('pagination', {
  template: `<li class="page-item">
    <a class="page-link" href="#" aria-label="Previous">
      <span aria-hidden="true">Previous</span>
    </a>
  </li>
  <li class="page-item"><a class="page-link" href="#">1</a></li>
  <li class="page-item"><a class="page-link" href="#">2</a></li>

  <li class="page-item">
    <a class="page-link" href="#" aria-label="Next">
      <span aria-hidden="true">Next</span>
    </a>
  </li>`
});

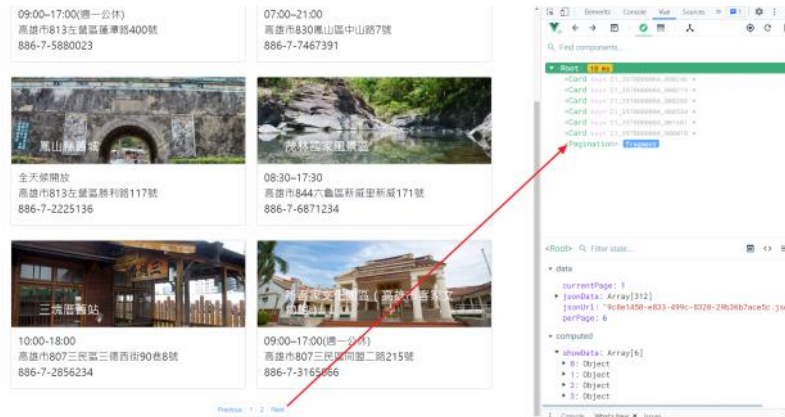
```



```

<div class="container" id="app">
  <div class="row pt-5">
    <card v-for="(item, key) in showData" :key="item.Id" :item="item"></card>
    <div class="d-flex justify-content-center mt-3">
      <nav aria-label="Page navigation example">
        <ul class="pagination pagination-sm">
          <pagination></pagination>
        </ul>
      </nav>
    </div>
  </div>
</div>

```



7. 計算分頁數並渲染

Math.ceil() 無條件進位

```

computed: {
  showData() { ...
  totalPage() {
    return Math.ceil(this.jsonData.length / this.perPage)
  }
}

```

傳入props的參數 total-page

```

<div class="container" id="app">
  <div class="row pt-5">
    <card v-for="(item, key) in showData" :key="item.Id" :item="item"></card>
    <div class="d-flex justify-content-center mt-3">
      <nav aria-label="Page navigation example">
        <ul class="pagination pagination-sm">
          <pagination :total-page="totalPage"></pagination>
        </ul>
      </nav>
    </div>
  </div>
</div>

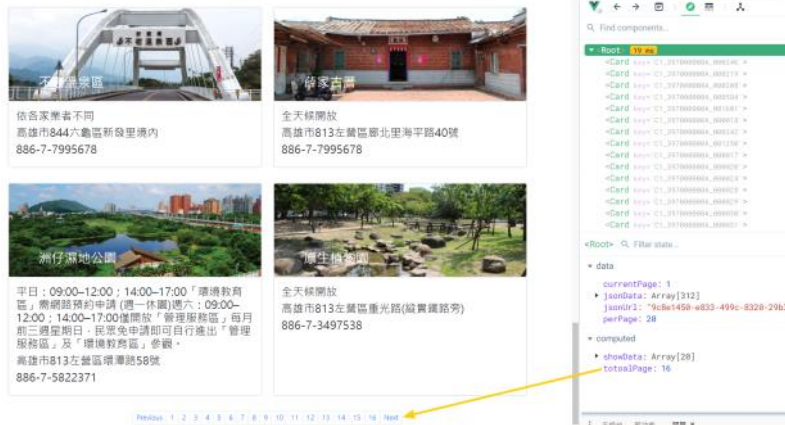
```

傳入component的參數 total-page需改成totalPage

```

App.component('pagination', {
  props: ['totalPage'],
  template: `
    <li class="page-item">
      <a class="page-link" href="#" aria-label="Previous">
        <span aria-hidden="true">Previous</span>
      </a>
    </li>
    <li class="page-item" v-for="(page, key) in totalPage">
      <a class="page-link" href="#">{{page}}</a>
    </li>
    <li class="page-item">
      <a class="page-link" href="#" aria-label="Next">
        <span aria-hidden="true">Next</span>
      </a>
    </li>
  `
});
App.mount('#app');

```



8. component 事件向外傳遞

Component 監聽事件

`@click.prevent="changePage(page)"`

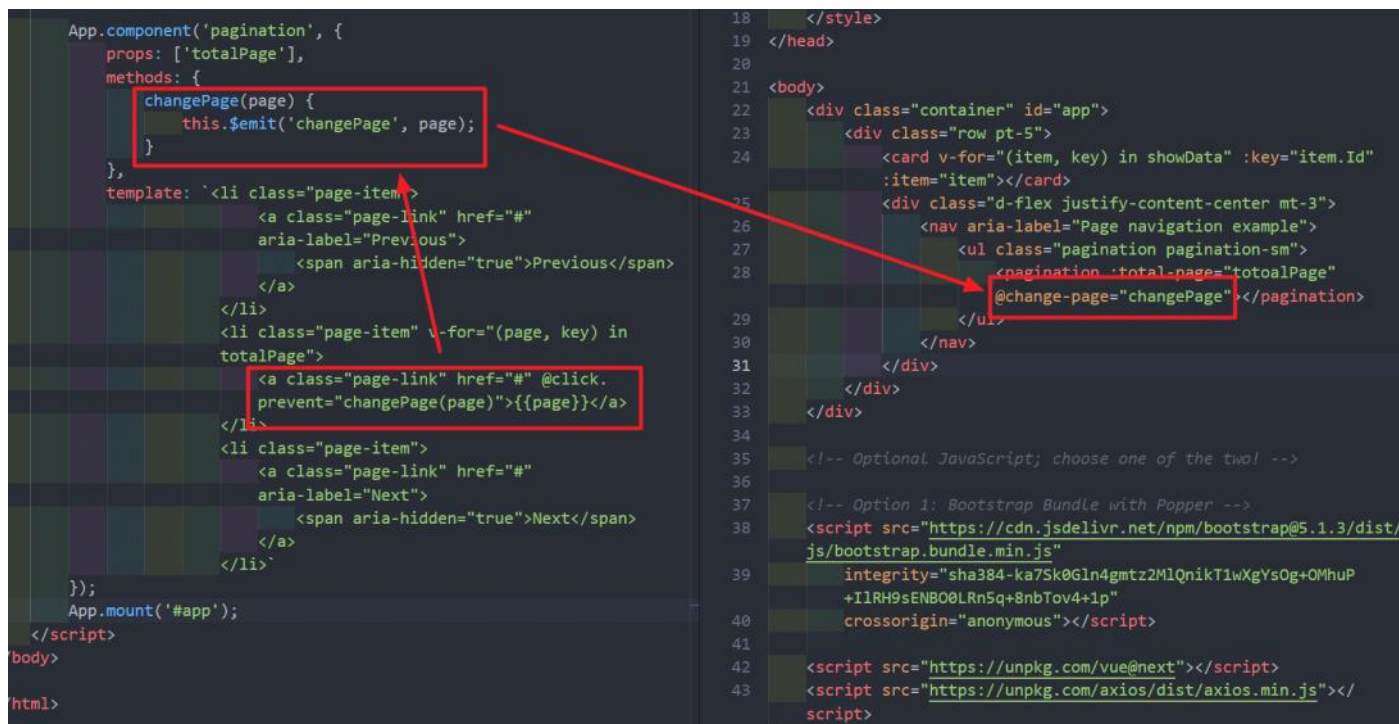
使用\$emit 向外傳遞事件與參數

```
changePage(page) {
  this.$emit('changePage', page);
}
```

外部 `@change-page` 接收 `this.$emit('changePage', page)` 傳入的監聽事件，由於命名間有大寫因此前面需加 '-'，觸發 `changePage` 事件。因為 `$emit` 已有傳遞參數，此處只需要呼叫 `changePage` 即可！

```
<pagination :total-page="totalPage" @change-page="changePage">
</pagination>
```

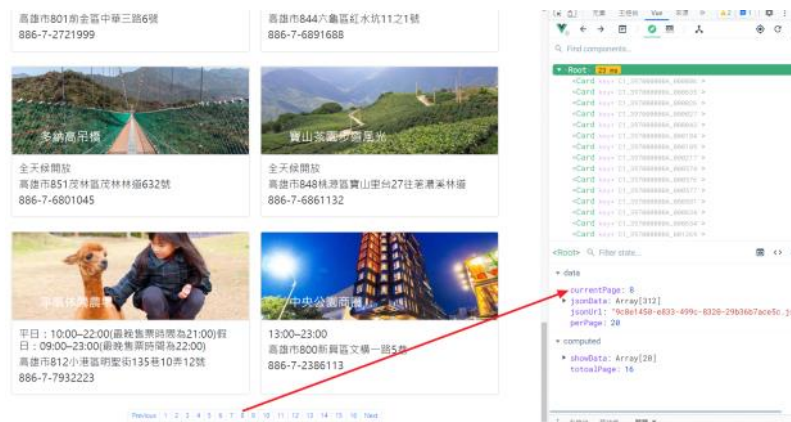
```
changePage(page) {
  this.currentPage = page;
}
```



```

methods: {
  //取得高雄旅遊網api資料
  getData() { ...
  changePage(page) {
    this.currentPage = page;
  }
},

```



9. 傳入 now-page並顯示目前頁面與上下頁功能

```

<div class="container" id="app">
  <div class="row">
    <card v-for="(item, key) in showdata" :key="item.Id" :item="item"></card>
    <div class="d-flex justify-content-center">
      <nav aria-label="Page navigation example">
        <ul class="pagination pagination-sm">
          <pagination :total-page="pageTotal" :now-page="currentPage"
            @change-page="changePage"></pagination>
        </ul>
      </nav>
    </div>
  </div>
</div>

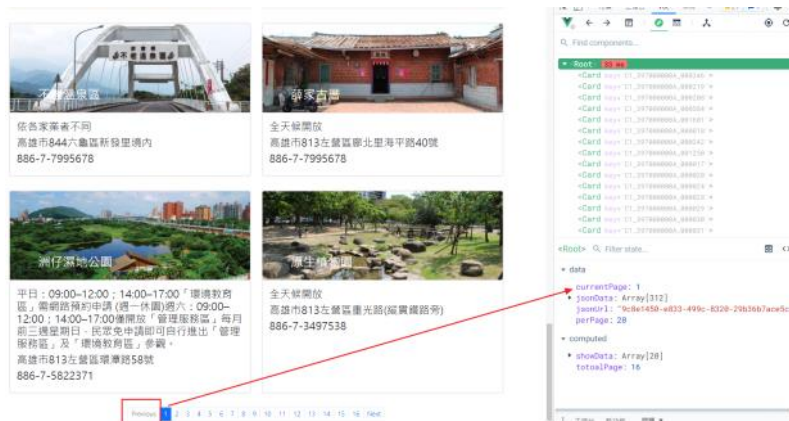
```

```

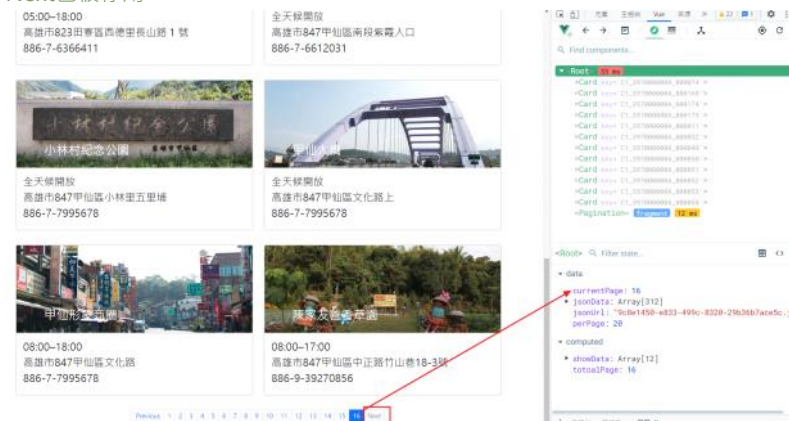
App.component('pagination', {
  props: ['totalPage', 'nowPage'],
  methods: {
    changePage(page) {
      this.$emit('changePage', page);
    }
  },
  template: `<li class="page-item" :class="{ 'disabled' : nowPage == 1}">
    <a class="page-link" href="#" aria-label="Previous" @click.prevent="changePage(nowPage-1)">
      <span aria-hidden="true">Previous</span>
    </a>
  </li>
  <li class="page-item" v-for="(page, key) in totalPage" :class="{ 'active' : page == nowPage}">
    <a class="page-link" href="#" @click.prevent="changePage(page)">{{page}}</a>
  </li>
  <li class="page-item" :class="{ 'disabled' : nowPage == totalPage}">
    <a class="page-link" href="#" aria-label="Next" @click.prevent="changePage(nowPage+1)">
      <span aria-hidden="true">Next</span>
    </a>
  </li>`
});
App.mount('#app');

```

Previous已被停用



Next已被停用



10. 完成後可以發現主要的html語法變得很簡短! 主要功能都是以Vue相關工具完成!

