



HTML、CSS 與 JavaScript 是網頁設計最核心、也最基礎的技術，無論您是想從頭開發一個網頁或改寫既有的網頁，這三種技術都是必學的基本功。此外，我們還會介紹響應式網頁設計 (RWD)、Bootstrap、Vue.js、jQuery 等進階的技術，幫助您更有效率地開發網頁。

隨著 ChatGPT、Gemini、Copilot、Perplexity、Claude 等 AI 助理的出現，程式設計這項技能開始有了轉變，傳統的學習方式是以語法為導向，而 AI 時代的學習方式是以解決問題為核心，當人們面對問題時，首先要思考的是如何解決問題，接著才是撰寫程式碼，然後測試程式碼。

因此，程式設計除了要懂得語法，還要培養解決問題的邏輯思維，才能有效率地跟 AI 助理溝通，生成符合需求的程式碼，然後進一步審閱並測試程式碼，確保程式碼是經過完整測試、正確無誤、安全且有效率的。

在本書中，我們會在章節的開頭講解重要的語法與主題，然後在結尾的地方透過「 ChatGPT 程式助理新思維」專欄，示範如何與 AI 助理協同作業，讓初學者不再糾結於繁瑣的語法，快速寫出正確、高效、有用的程式！

本書內容

- ✓ **Part 1 — HTML**：HTML 用來定義網頁的內容，開發各種網頁應用程式。在本篇中，我們會介紹常用的 HTML 元素，例如文件結構、資料編輯與格式化、影音多媒體、表格、表單等。
- ✓ **Part 2 — CSS**：CSS 用來定義網頁的外觀，包括編排、顯示、格式化及特殊效果。在本篇中，我們會介紹常用的 CSS 屬性，例如色彩、字型、文字、清單、表格、背景、濾鏡、漸層、變形、轉場、動畫、Box Model、定位方式、版面設計、媒體查詢等，並以實例示範響應式網頁設計 (RWD)。
- ✓ **Part 3 — JavaScript**：JavaScript 用來定義網頁的行為。在本篇中，我們會介紹 JavaScript 的基本語法，例如型別、變數、常數、運算子、流程控制、函式、物件等，還會介紹 JavaScript 在用戶端的應用，包括文件物件模型 (DOM)、事件處理等。

✔ Part 4 — 前端框架：在本篇中，我們會介紹下列幾種技術：

- **Bootstrap**：這是一個 HTML、CSS 與 JavaScript 框架，用來開發響應式、行動優先的網頁，使用者無須撰寫 CSS 或 JavaScript 程式碼，就可以輕鬆設計出響應式網頁。
- **Vue.js**：這是一個用來建立使用者介面的 JavaScript 框架，提供 API 讓 Web 開發人員進行資料繫結及操作網頁上的元素，解決畫面顯示與資料狀態同步的問題。
- **jQuery**：這是一個快速、輕巧、功能強大的 JavaScript 函式庫，提供 API 讓操作 HTML 文件、選擇 HTML 元素、處理事件、建立特效、使用 Ajax 等動作變得更簡單。

排版慣例

本書在條列 HTML 元素、CSS 屬性與 JavaScript 語法時，遵循下列排版慣例：

- ✔ 斜體字表示網頁設計人員輸入的屬性值、敘述或名稱，例如 `src="url"` 的 *url* 表示自行輸入的網址。
- ✔ 中括號 `[]` 表示可以省略不寫，例如 `function` 函式名稱 (`[ 參數 ]`) 的 `[ 參數 ]` 表示函式的參數可以有，也可以沒有。
- ✔ 垂直線 `|` 用來隔開替代選項，例如 `caption-side: top | botom` 表示 `caption-side` (表格標題位置) 屬性的值可以是 `top` 或 `bottom`。

聯絡方式

如果您有建議，或授課老師需要 PowerPoint 教學投影片，歡迎與我們聯絡：基峯資訊網站 <https://www.gotop.com.tw/>；國內學校業務處電話—台北 (02)2788-2408、台中 (04)2452-7051、高雄 (07)384-7699。



1 準備如下的表格資料「首都與國花.xlsx」或「首都與國花.docx」。

國家	首都	國花
芬蘭	赫爾辛基	鈴蘭
德國	柏林	矢車菊
荷蘭	阿姆斯特丹	鬱金香
波蘭	華沙	三色堇
奧地利	維也納	雪絨花
義大利	羅馬	雛菊
希臘	雅典	油橄欖
西班牙	馬德里	香石竹
法國	巴黎	香根鳶尾
英國	倫敦	玫瑰

改寫成網頁，暫時不要設定 CSS 樣式，得到如下回答。



3 將程式碼複製到編輯器，加以檢視、編輯、儲存與瀏覽。

✓ \Ch04\flower.html

```
<!DOCTYPE html>
<html lang="zh-Hant">
  <head>
    <meta charset="UTF-8">
    <title> 首都與國花 </title>
  </head>
  <body>
    <table border="1">
      <thead>
        <tr>
          <th> 國家 </th><th> 首都 </th><th> 國花 </th>
        </tr>
      </thead>
      <tbody>
        <tr><td> 芬蘭 </td><td> 赫爾辛基 </td><td> 鈴蘭 </td></tr>
        <tr><td> 德國 </td><td> 柏林 </td><td> 矢車菊 </td></tr>
        <tr><td> 荷蘭 </td><td> 阿姆斯特丹 </td><td> 鬱金香 </td></tr>
        <tr><td> 波蘭 </td><td> 華沙 </td><td> 三色堇 </td></tr>
        <tr><td> 奧地利 </td><td> 維也納 </td><td> 雪絨花 </td></tr>
        <tr><td> 義大利 </td><td> 羅馬 </td><td> 雛菊 </td></tr>
        <tr><td> 希臘 </td><td> 雅典 </td><td> 油橄欖 </td></tr>
        <tr><td> 西班牙 </td><td> 馬德里 </td><td> 香石竹 </td></tr>
        <tr><td> 法國 </td><td> 巴黎 </td><td> 香根鳶尾 </td></tr>
        <tr><td> 英國 </td><td> 倫敦 </td><td> 玫瑰 </td></tr>
      </tbody>
    </table>
  </body>
</html>
```

國家	首都	國花
芬蘭	赫爾辛基	鈴蘭
德國	柏林	矢車菊
荷蘭	阿姆斯特丹	鬱金香
波蘭	華沙	三色堇
奧地利	維也納	雪絨花
義大利	羅馬	雛菊
希臘	雅典	油橄欖
西班牙	馬德里	香石竹
法國	巴黎	香根鳶尾
英國	倫敦	玫瑰

為了方便檢查，我們要求暫時不要設定 CSS 樣式，因此，ChatGPT 生成的是純 HTML 網頁。在確認表格資料和程式碼都沒有問題後，就可以要求美化表格，例如「**根據這個純 HTML 網頁加上樣式設定，讓表格呈現專業的外觀**」。若對於表格的配色有意見，可以把色系告訴 ChatGPT；若要新增或刪除表格資料，可以直接下指令，例如「**刪除這個網頁中關於芬蘭的那列資料**」。

9-3 手刻響應式網頁 (圖庫網站)

在本節中，我們將透過下面的例子示範響應式網頁設計。由於手機、平板電腦與 PC 常見的可視區域寬度是 320 ~ 480px、768px 以上、1024px 以上，因此，我們採取 500px 和 768px 兩個斷點，並將網頁的最大寬度設定為 960px。

圖 1 為小尺寸裝置的瀏覽結果，當可視區域的寬度 $\leq 500\text{px}$ 時 (手機版)，主要內容會顯示單欄；圖 2 為中尺寸裝置的瀏覽結果，當可視區域的寬度介於 501px ~ 768px 時 (平板電腦版)，主要內容會顯示兩欄。

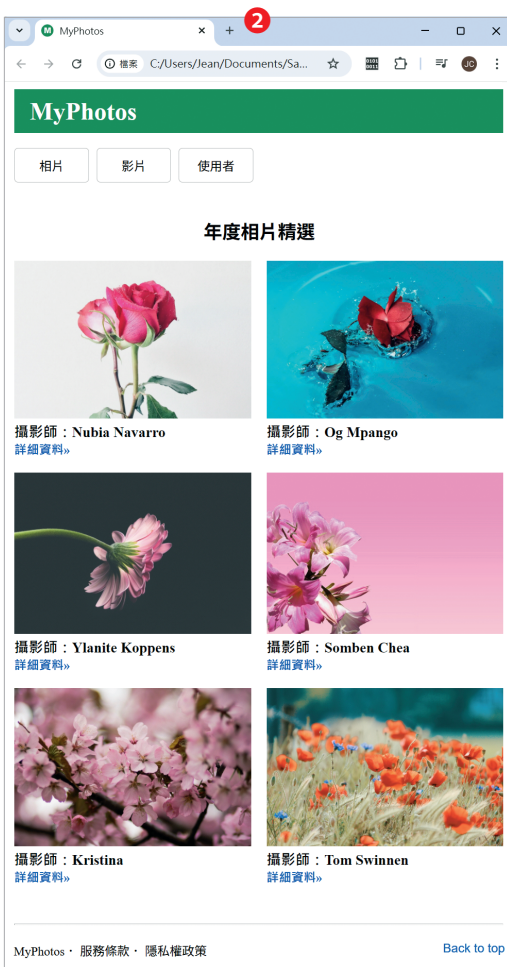
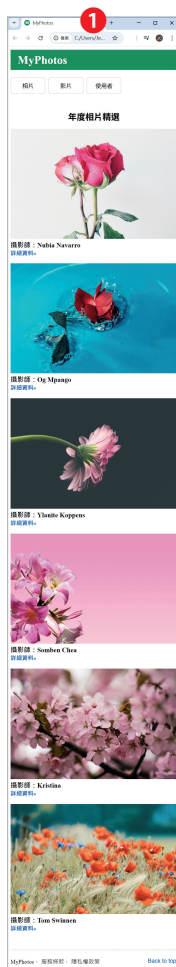
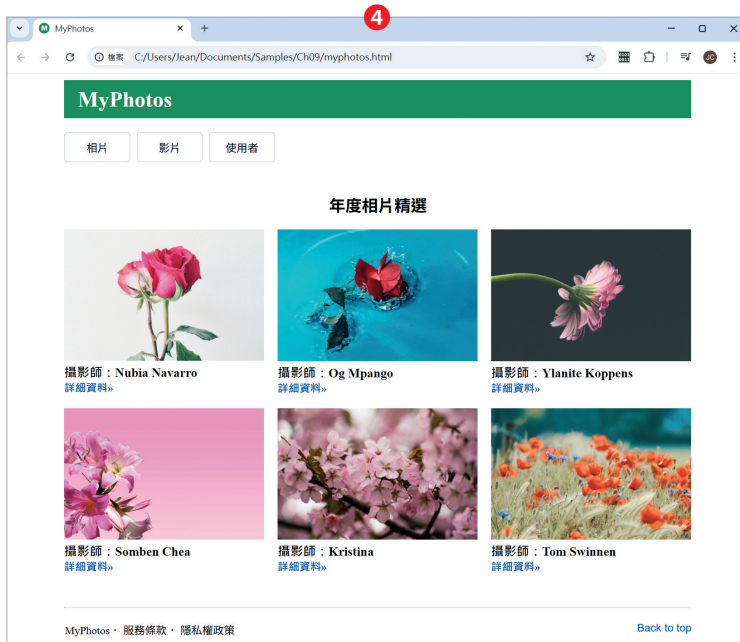
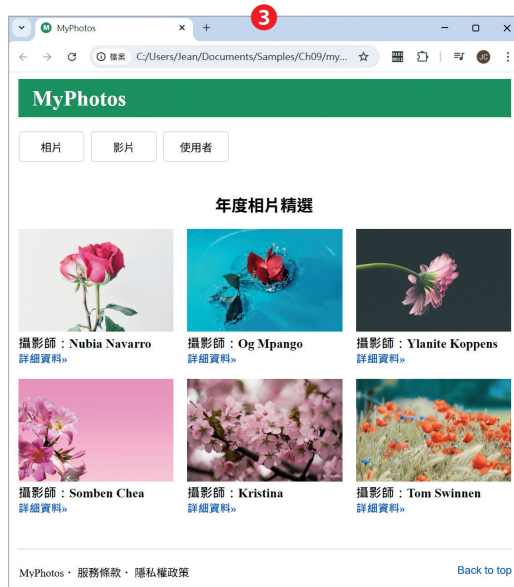


圖 3 為大尺寸裝置的瀏覽結果，當可視區域的寬度 $\geq 769\text{px}$ 時 (PC 版)，主要內容會顯示三欄；圖 4 亦為大尺寸裝置的瀏覽結果，當可視區域的寬度大於 960px 時，網頁內容會維持 960px ，兩側顯示空白。



10-3-7 陣列 (array)

陣列 (array) 可以用來儲存多個資料，這些資料叫做**元素** (element)，每個元素有各自的**索引** (index) 與**值** (value)。

索引可以用來識別元素，例如第 1 個元素的索引為 0，第 2 個元素的索引為 1，...，第 n 個元素的索引為 n - 1。當陣列最多儲存 n 個元素時，表示它的**長度** (length) 為 n。

例如下面的敘述是建立一個陣列並指派給變數 A：

let A = [10, 20, 30];

- ❶ 陣列的名稱。
- ❷ 陣列的前後以中括號括起來。
- ❸ 包含 10、20、30 三個元素，中間以逗號隔開。我們可以透過陣列的名稱與索引來存取元素，例如 A[0]、A[1]、A[2] 分別代表 10、20、30。

元素	值
A[0]	10
A[1]	20
A[2]	30

陣列裡面也可以儲存其它陣列，形成**巢狀陣列** (nested array)，例如下面的敘述是建立一個巢狀陣列並指派給變數 B：

let B = [10, [21, 22], 30];

- ❶ 巢狀陣列的名稱。
- ❷ 第二個元素是另一個陣列，我們可以透過陣列的名稱與兩個索引來存取元素，例如 B[1][0]、B[1][1] 分別代表 21、22。

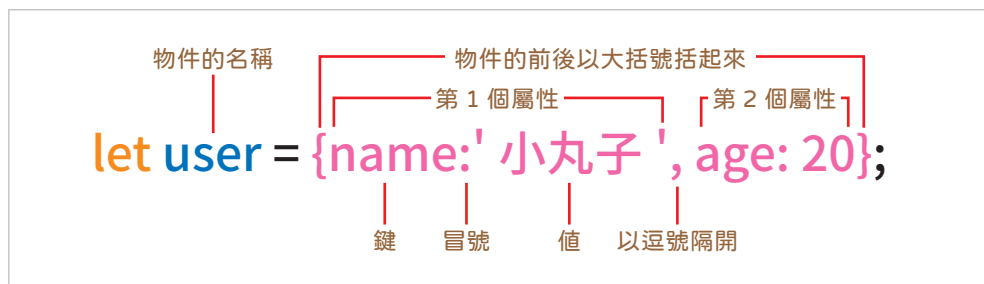
元素	值
B[0]	10
B[1]	[21, 22]
B[1][0]	21
B[1][1]	22
B[2]	30

10-3-8 物件 (object)

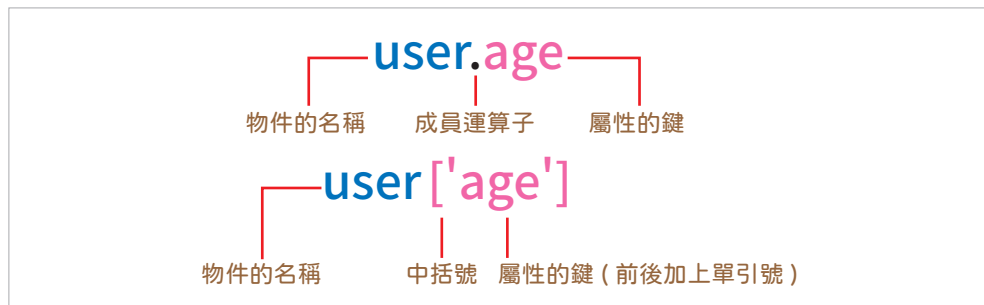
JavaScript 的物件 (object) 是一種關聯陣列 (associative array)，它和一般陣列的差別如下：

- ✓ 陣列所儲存的資料稱為**元素** (element)，而物件所儲存的資料稱為**屬性** (property)，屬性除了可以是數值、字串、布林等資料，也可以是函式，這種儲存了函式的屬性又稱為**方法** (method)。
- ✓ 陣列是使用**索引** (index) 來識別元素，而物件是使用**鍵** (key) 來識別屬性，索引是數字，而鍵是字串。事實上，物件的屬性就是一個**鍵 / 值對** (key/value pair)，分別代表屬性的名稱與值。

例如下面的敘述是建立一個物件並指派給變數 user：

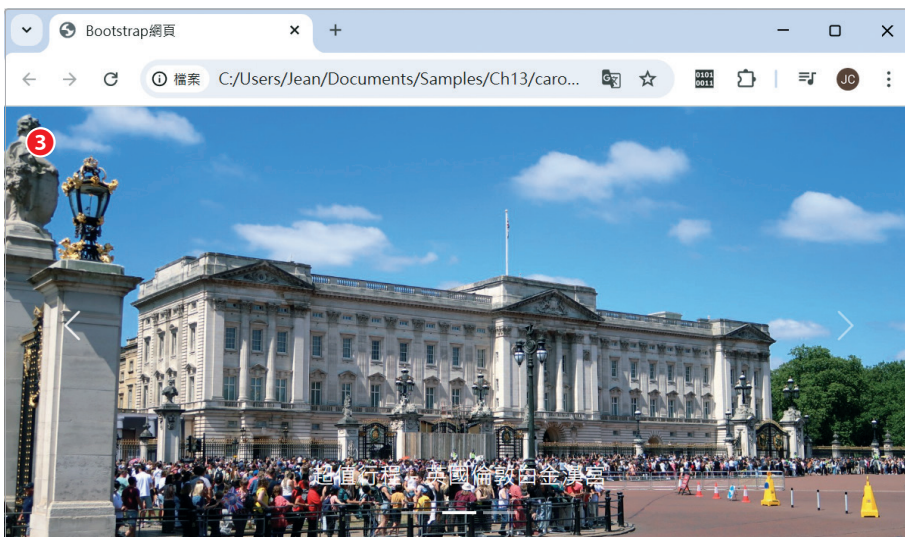
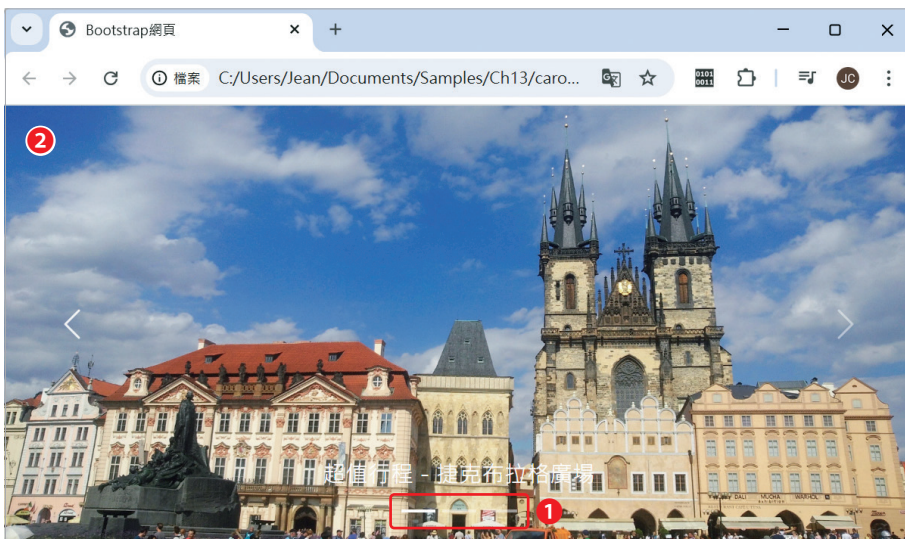


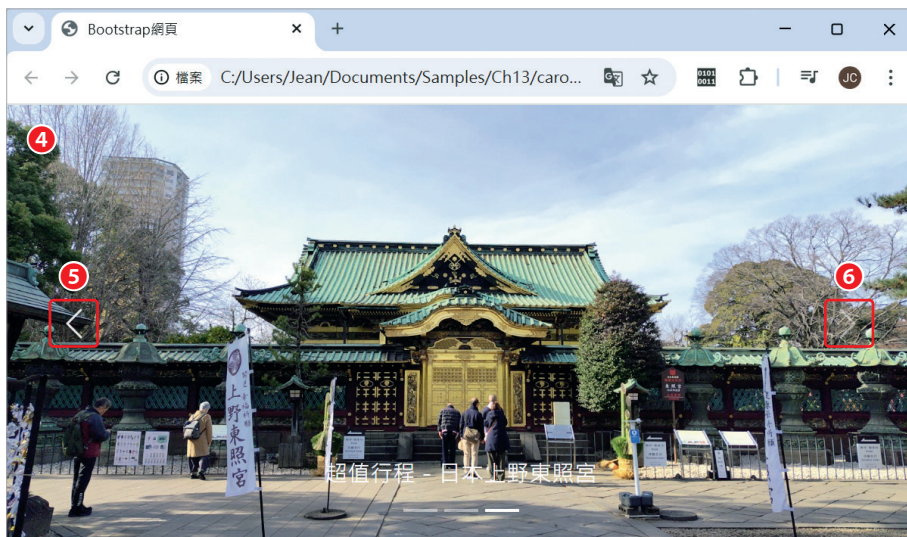
我們可以使用**成員運算子** (.) 或**中括號表示法** 存取物件的屬性，例如下面兩個寫法均會傳回 age 屬性的值，也就是 20。有關如何操作物件的屬性與方法，第 11 章有進一步的說明。



13-12 輪播 (Carousel)

輪播 (Carousel) 元件用來在網頁上循環播放多張圖片、影片或其它類型的內容，就像幻燈片一樣。下面是一個例子，它會循環播放三張圖片。





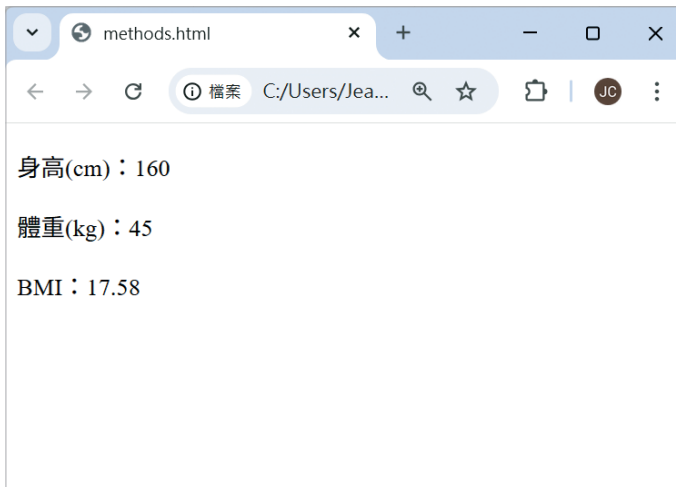
- | | |
|-------------------|-------------------|
| ❶ 輪播指示 (點選圖片) | ❷ 輪播內容 (第一張圖片與標題) |
| ❸ 輪播內容 (第二張圖片與標題) | ❹ 輪播內容 (第三張圖片與標題) |
| ❺ 輪播控制 (上一個) | ❻ 輪播控制 (下一個) |

輪播元件是以 CSS 3D 變形與少量的 JavaScript 所建立，可以和一系列圖片、文字或自訂標記一起使用，包含下列幾個部分：

- ✓ **輪播指示** (carousel-indicators)：由 `<div>` 和 `<button>` 元素所組成，用來根據圖片的張數在圖片下方顯示橫線以供使用者點選，此例有三張圖片，所以會顯示三個橫線。
- ✓ **輪播內容** (carousel-inner)：由多個層次的 `<div>` 元素所組成，用來顯示內容，此例會顯示圖片與標題。
- ✓ **輪播控制** (carousel-control)：由兩個 `<button>` 元素所組成，用來顯示「上一個」/「下一個」控制按鈕。

- ✔ 05：在 Mustache 標籤中呼叫 BMI() 方法，所謂「方法」其實就是物件裡面的函式，正因為是函式呼叫，所以必須加上小括號，若函式有參數，寫在小括號裡面即可。
- ✔ 16 ~ 20：在 methods 屬性中定義一個名稱為 BMI 的方法，傳回值為體重 (公斤) 除以身高 (公尺) 的平方，呼叫 toFixed(2) 方法可以取到小數點後面二位數，瀏覽結果如下圖。

請注意，第 18 行有一個 **this** 關鍵字，指的是此物件本身，也就是目前的應用程式實體，我們可以透過 this.weight 和 this.height 存取 data 屬性中的 weight 和 height；同理，若要在相同應用程式實體的其它方法中呼叫 BMI() 方法，只要寫成 this.BMI() 即可。



14-4-2 computed 屬性

computed 屬性可以用來將一些程式碼執行完畢的結果當作取出的資料值，它同樣是以函式的形式呈現。

舉例來說，我們可以換用 computed 屬性將前一節的例子改寫成如下，瀏覽結果是相同的。從這兩個例子可以看出，methods 與 computed 屬性都能夠用來將一些運算過程包裝起來以供重複使用。

✓ \Ch14\computed.html

```
<body>
  <div id="app">
    <p> 身高 (cm) : {{ height }}</p>
    <p> 體重 (kg) : {{ weight }}</p>
    <p> BMI : {{ BMI }}</p>
  </div>
  <script src="computed.js"></script>
</body>
```

✓ \Ch14\computed.js

```
Vue.createApp({
  data() {
    return {
      height: 160,
      weight: 45
    }
  },
  computed: {
    BMI() {
      return (this.weight / (this.height / 100) ** 2).toFixed(2);
    }
  }
}).mount('#app');
```



- 1 在 Mustache 標籤中存取 BMI 的值，注意此處沒有小括號
- 2 在 computed 屬性中定義 BMI 的值

