關於本書

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**、**jQuery**、**Vue.js** 等進階的技術，幫助您更有效率地開發網頁。

在 **ChatGPT** 橫空出世後，有不少人驚覺「寫程式」即將被 AI 工具取代，沒錯，AI 工具確實能夠寫程式，但這並不表示您就不用學程式設計，而是程式設計師必須要進化為 AI 工具的程式審查員或教 AI 學習的老師。

換句話說，您必須具備程式設計能力，才有辦法跟 AI 工具溝通，讓它寫出您需要的程式碼，也才有辦法閱讀或審查 AI 工具所生成的程式碼，確保程式碼是正確的、有效率的、經過完整測試的，而想要練就扎實的程式功力，您所需要的正是一本好書。

在本書的一開始，我們會先告訴您如何使用 ChatGPT 學習網頁設計並撰寫網頁程式，之所以將 ChatGPT 放在最前面的章節，主要是因為這些技巧都可以運用到目前的網頁技術，與其分散到各個章節，倒不如集中在一個章節，比較能夠有系統地學習。

本書內容

- **第 0 章－ ChatGPT**：在本章中，我們會介紹如何使用 ChatGPT 學習網頁設計，例如撰寫網頁程式、解讀網頁程式、加註解、除錯、查詢語法、尋求技術支援、出題練習、解題、與其它程式語言做轉換等。
- **Part 1 － HTML**：HTML 可以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 HTML 常用的元素，例如文件結構、資料編輯與格式化、嵌入內容、表格、表單等。
- **Part 2 － CSS**：CSS 可以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 CSS 常用的屬性，例如色彩、字型、文字、清單、Box Model、定位方式、背景、漸層、濾鏡、表格、Flexbox Layout、Grid Layout、變形、轉場、動畫、媒體查詢等。

- **Part 3 － JavaScript**：JavaScript 可以用來定義網頁的行為，在本篇中，我們會介紹 JavaScript 的基本語法，包括型別、變數、常數、運算子、流程控制、函式、物件等，還會介紹 JavaScript 在瀏覽器端的應用，也就是如何利用 JavaScript 讓靜態網頁具有動態效果，包括文件物件模型 (DOM)、瀏覽器物件模型 (BOM)、事件處理等。

- **Part 4 －網頁前端框架**：在本篇中，我們會介紹下列幾種技術：

  - **Bootstrap**：這是很受歡迎的 HTML、CSS 與 JavaScript 框架，用來開發響應式 (responsive)、行動優先 (mobile first) 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。

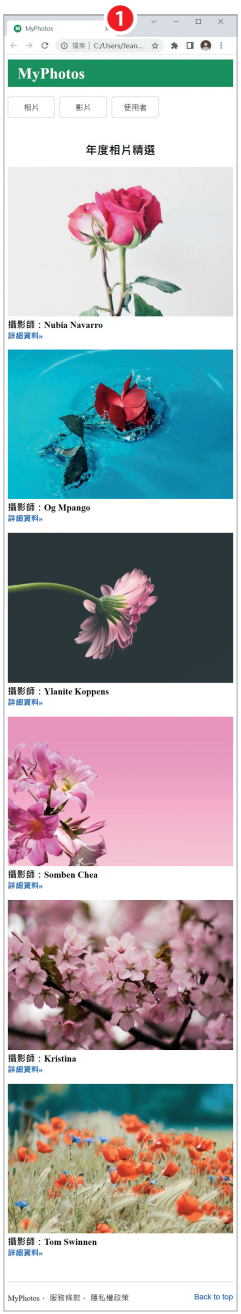
  - **jQuery**：這是一個快速、輕巧、功能強大的 JavaScript 函式庫，透過它所提供的 API，可以讓諸如操作 HTML 文件、選擇 HTML 元素、處理事件、建立特效、使用 Ajax 技術等動作變得更簡單。

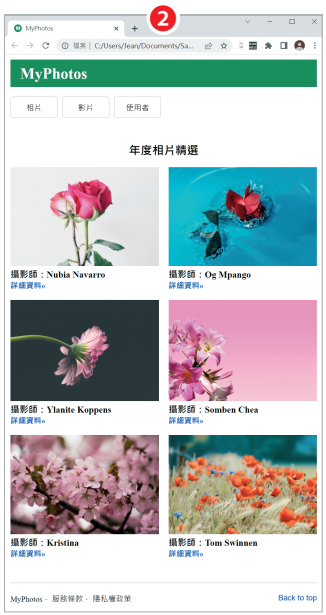
  - **Vue.js**：除了 Bootstrap 和 jQuery，還有許多應用於 JavaScript 程式開發的函式庫與框架，例如 Vue.js、React、Angular、Backbone.js 等，其中 Vue.js 是一個 JavaScript 函式庫，提供 API 讓 Web 開發人員進行資料繫結及操作網頁上的元素，解決畫面顯示與資料狀態同步的問題。由於 Vue.js 簡單易學、容易導入並具有高度的擴充性，所以我們也將它納入本書一併做介紹。

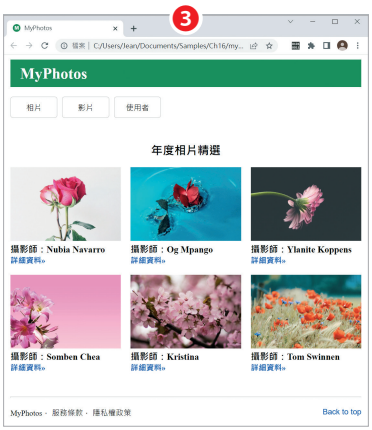
此外，我們還會介紹**響應式網頁設計** (RWD，Responsive Web Design)，這是一種網頁設計方式，目的是根據使用者的瀏覽器環境 (例如寬度或方向等)，自動調整網頁的版面配置，以提供最佳的顯示結果，換句話說，只要設計單一版本的網頁，就能完整顯示在 PC、平板電腦、智慧型手機等裝置。

為了讓您體驗如何將各種語法活用到實際的網頁設計，我們設計了**三個大範例－「圖庫網站」、「旅遊網站」和「部落格網站」**，其中第一個範例是使用 HTML 和 CSS 手刻響應式網頁，而第二、三個範例是使用 HTML、CSS 和 Bootstrap 開發響應式網頁，這些範例均相當精美，對於設計人員開發網頁、老師設計教學範例、學生製作專題或參加競賽都極具參考價值。

本書範例網站 01　手刻響應式網頁－圖庫網站



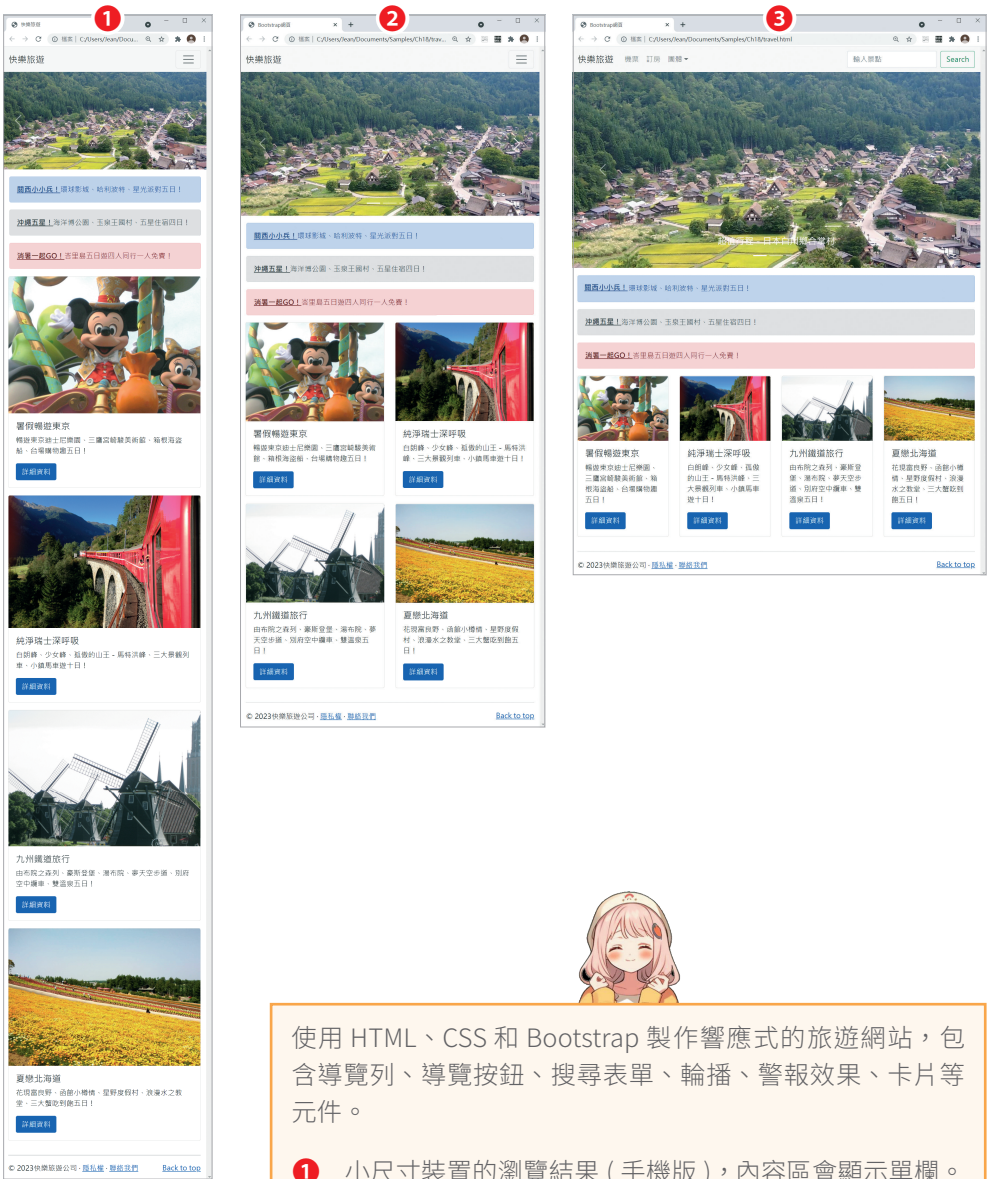




使用 HTML 和 CSS 手刻響應式的圖庫網站，包含斷點設計、版面設計、彈性盒子版面 (Flexible Box Layout)、格線版面 (Grid Layout)、網站圖示等技巧。

1. 小尺寸裝置的瀏覽結果 (手機版)，內容區會顯示單欄。
2. 中尺寸裝置的瀏覽結果 (平板電腦版)，內容區會顯示兩欄。
3. 大尺寸裝置的瀏覽結果 (PC 版)，內容區會顯示三欄。

本書範例網站 02　Bootstrap 響應式網頁－旅遊網站



使用 HTML、CSS 和 Bootstrap 製作響應式的旅遊網站，包含導覽列、導覽按鈕、搜尋表單、輪播、警報效果、卡片等元件。

1. 小尺寸裝置的瀏覽結果 (手機版)，內容區會顯示單欄。
2. 中尺寸裝置的瀏覽結果 (平板電腦版)，內容區會顯示兩欄。
3. 大尺寸裝置的瀏覽結果 (PC 版)，內容區會顯示四欄。

本書範例網站 03　Bootstrap 響應式網頁－部落格網站

使用 HTML、CSS 和 Bootstrap 製作響應式的部落格網站，包含頁首、導覽列、搜尋圖示、文章、側邊欄、頁尾等區塊。

1. 中小尺寸裝置的瀏覽結果，內容區會顯示單欄，先顯示文章，再顯示側邊欄。
2. 大尺寸裝置的瀏覽結果，內容區會顯示雙欄，包含文章與側邊欄。

CHAPTER

00

網頁設計 ×ChatGPT

The Complete Guide to Web Design

0-4 撰寫網頁程式與除錯

我們可以請 ChatGPT 撰寫網頁程式、解讀網頁程式的意義、幫忙加上註解或進行除錯，以下有進一步的說明。

0-4-1 撰寫指定用途的網頁程式

當您要請 ChatGPT 撰寫網頁程式時，請詳細描述程式的用途，這樣它所撰寫出來的程式就會更符合要求，例如「請撰寫一個網頁，裡面有一個連結到 Google 台灣的超連結」、「請撰寫一個網頁，裡面以表格列出春夏秋冬的中英對照」、「請給我實現氣泡排序法的 JavaScript 程式」、「請給我使用 for 迴圈實現九九乘法表的 JavaScript 程式」、「請撰寫一個網頁，裡面有一個標題 1 格式的字串 "Hello!"，當指標移到該字串時，字串會變成紅色，當指標離開該字串時，字串會恢復成預設的顏色」等。

下面是在 ChatGPT 輸入「**請給我計算整數 1 加到 10 的總和的 JavaScript 程式**」所得到的回答，裡面除了有程式碼，還有相關的說明，若要複製程式碼，只要點取程式碼方塊右上角的 **[Copy code]** 即可。

0-4-4 解決網頁程式的錯誤 (除錯)

當您遇到網頁程式執行錯誤時，可以向 ChatGPT 提問，讓它幫助您解決錯誤，也就是**除錯** (debug)，例如「為什麼這個程式執行錯誤？」、「我該如何解決這個程式的錯誤？」等。

下面是在 ChatGPT 輸入「**這段程式碼有什麼錯誤？**」，然後附上一段程式碼，ChatGPT 會先確認程式碼的意圖，接著根據意圖找出錯誤，解釋這樣的錯誤應該如何修正，然後提供修正完畢的程式碼。

JE

這段程式碼有什麼錯誤？

```
let sum = 0;
for (let i = 1; i <= 10; i++) {
 sum + i;
}
console.log(sum);
```

這段程式碼的錯誤在於在 `for` 迴圈內，應該將 `sum` 與 `i` 相加後的結果重新賦值給 `sum`。但是，目前的代碼中只有 `sum + i` 這個表達式，它並未更新 `sum` 的值。應該使用 `+=` 運算符來修正這個問題。

以下是修正後的程式碼：

javascript Copy code

```javascript
let sum = 0;
for (let i = 1; i <= 10; i++) {
  sum += i; // 使用 '+=' 運算符將 sum 與 i 相加後的結果重新賦值給 sum
}
console.log(sum);
```

在修正後的程式碼中，`sum += i` 將 `sum` 與 `i` 相加後的結果重新賦值給 `sum`。這樣，程式碼將正確地計算 1 到 10 之間所有整數的和並將結果打印到控制台。

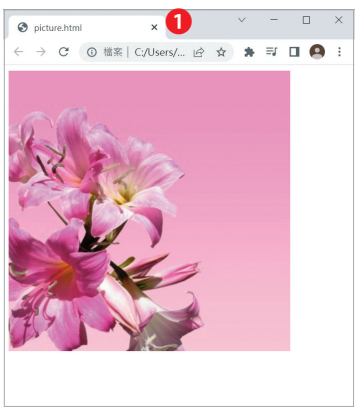
4-5 嵌入不同的圖片 – <picture> 元素

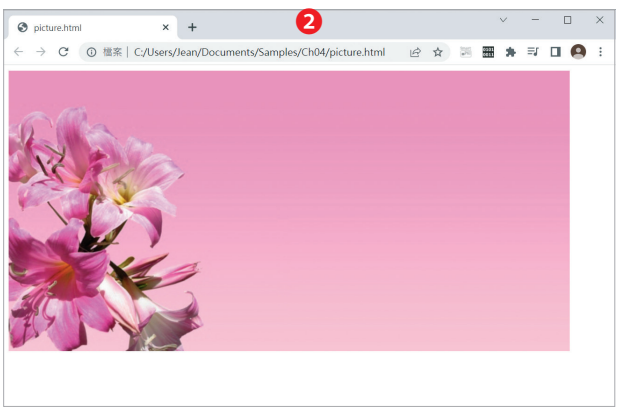
<picture> 元素可以用來針對不同的裝置或顯示畫面嵌入不同的圖片，其屬性有第 2-1 節所介紹的全域屬性。下面是一個例子，它會以 800 像素為分界，顯示兩張不同尺寸、不同內容的圖片。

\Ch04\picture.html

```
01 : <body>
02 :   <picture>
03 :     <source media="(max-width: 799px)" srcset="flower-small.jpg">
04 :     <source media="(min-width: 800px)" srcset="flower-large.jpg">
05 :     <img src="flower-large.jpg" alt=" 美麗的花兒 ">
06 :   </picture>
07 : </body>
```

- 03：使用 <source> 元素設定當可視區域的寬度≦ 799 像素時，就載入 flower-small.jpg，如圖 ❶，這張圖片的漸層背景比較小。
- 04：使用 <source> 元素設定當可視區域的寬度≧ 800 像素，就載入 flower-large.jpg，如圖 ❷，這張圖片的漸層背景比較大 。
- 05：若瀏覽器不支援 <source> 元素或選不到適合的圖檔，就載入 <img> 元素所指定的 flower-large.jpg。





11-2 彈性盒子版面

CSS3 的 Flexible Box Layout Module 提供了一組屬性可以用來製作**彈性盒子版面** (Flexible Box Layout)，簡稱為 **Flexbox**。

Flexbox 的排版原則很簡單，就是在 HTML 文件中建立一個稱為 **Flex Container** (彈性容器) 的父元素，然後在父元素中放入一個或多個稱為 **Flex Item** (彈性項目) 的子元素。

下面是一個例子，外層的 元素是父元素，用來做為 Flex Container，而內層的三個 元素是子元素，用來做為 Flex Item。

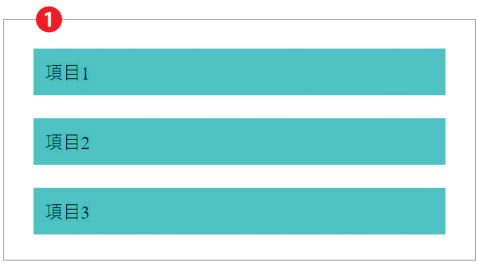
\Ch11\flex1.html

```
<div class="container">
  <div class="item">項目 1</div>
  <div class="item">項目 2</div>
  <div class="item">項目 3</div>
</div>
```

\Ch11\flex1.css

```
.container {
  display: flex;
}
.item {
  background: cyan;
  padding: 10px;
  margin: 10px;
}
```

在預設的情況下，父元素的顯示類型為 block，所以三個子元素的瀏覽結果如左下圖，呈現由上往下排列，而在將父元素的顯示類型設定為 flex 後，三個子元素的瀏覽結果如右下圖，呈現由左往右排列。



1 block 顯示類型會由上往下排列

2 flex 顯示類型會由左往右排列

14-3-7 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：

```
var 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：

```
var 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

18-6-3 滑鼠事件

滑鼠事件是一些與使用者操作滑鼠相關的事件，例如 click、dblclick、mousedown、mouseup、mouseenter、mouseleave、mouseover、mouseout、mousemove、mousewheel 等，其中 **mouseover / mouseout** 會在使用者將滑鼠移入 / 移出元素時觸發。下面是一個例子，剛開始網頁上會顯示圖片 girl1.png，當指標移到圖片時會變成 girl2.png，而當指標離開圖片時又會變成原來的 girl1.png。

\Ch18\mouseover.html

```
<body>
  <img id="fig" src="girl1.png" width="200">
  <script src="mouseover.js"></script>
</body>
```

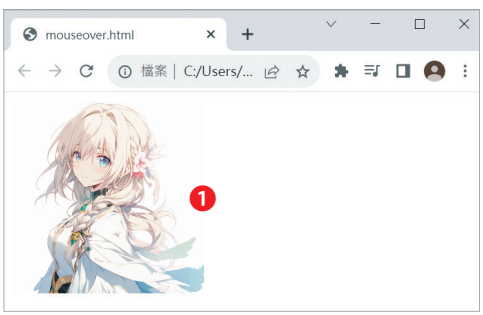
\Ch18\mouseover.js

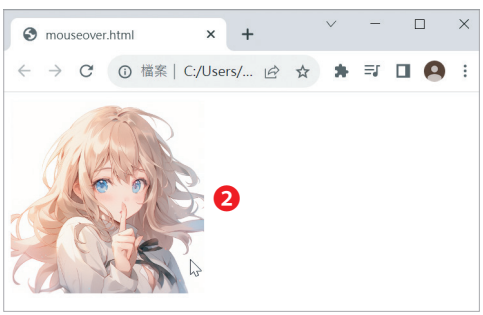
```
var fig = document.getElementById('fig');

fig.addEventListener('mouseover', function() {
  fig.src = 'girl2.png';
}, false);

fig.addEventListener('mouseout', function() {
  fig.src = 'girl1.png';
}, false);
```

1. 當指標離開圖片時會顯示 girl1.png
2. 當指標移到圖片時會顯示 girl2.png





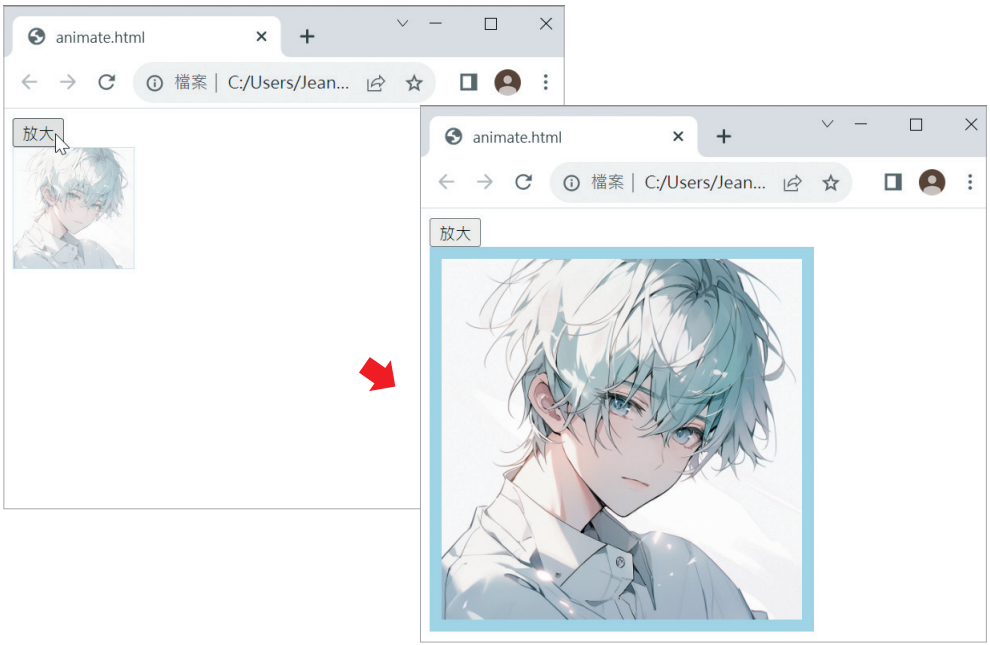
24-5-3 自訂動畫

jQuery 提供的 **.animate()** 方法可以針對元素的 CSS 屬性自訂動畫，其語法如下：

```
.animate(properties [, duration] [, easing] [, complete])
```

- *properties*：設定欲套用動畫的 CSS 屬性與值。
- *duration*：設定動畫的執行時間，預設值為 400 (毫秒)。
- *easing*：設定在動畫套用不同的變化速度，預設值為 swing (在中段會加速，在前段和後段則較慢)，亦可設定為 linear (等速)。
- *complete*：設定動畫結束時所要執行的函式。

下面是一個例子，當使用者按一下 [放大] 時，會在 1500 毫秒內將圖片從寬度 100px、透明度 0.5、框線寬度 1px 逐漸放大到寬度 300px、完全不透明、框線寬度 10px。



\Ch24\animate.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      img {
        width: 100px;
        opacity: 0.5;
        border: 1px solid lightblue;
      }
    </style>
  </head>
  <body>
    <button id="enlarge"> 放大 </button><br>
    <img src="boy.png">
    <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
    <script src="animate.js"></script>
  </body>
</html>
```

\Ch24\animate.js

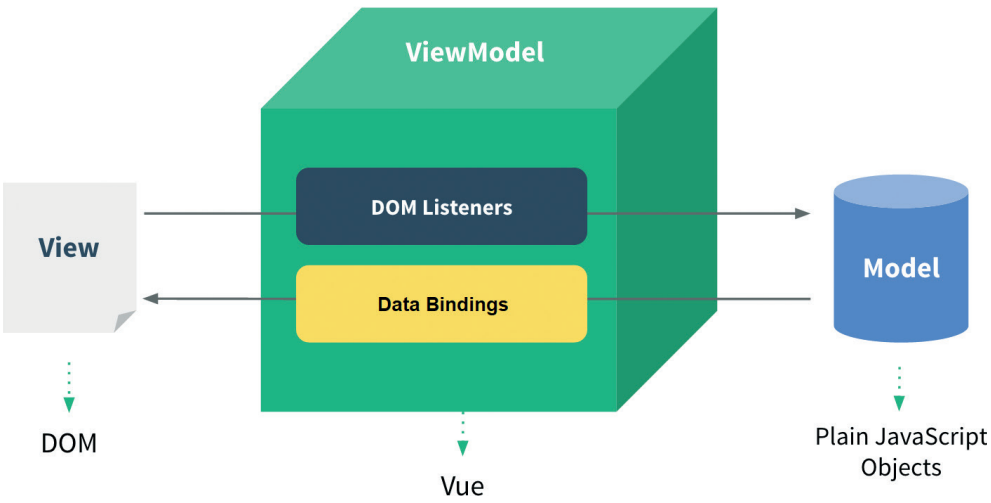
```
/* 令圖片在 1500 毫秒內逐漸放大到寬度 300px、完全不透明、框線寬度 10px */
$('#enlarge').on('click', function() {
  $('img').animate({
    width: '300px',
    opacity: 1,
    borderWidth: '10px'
  }, 1500);
});
```

Vue.js 與 MVVM

MVVM (Model-View-ViewModel) 是一種軟體架構模式，有助於將軟體開發的商業邏輯與畫面顯示分隔開來。MVVM 是由下列三個部分所組成：

- **Model**：資料狀態 (資料層)，負責管理資料。
- **View**：畫面顯示 (視圖層)，也就是使用者所看到的網頁畫面。
- **ViewModel**：資料連結器，做為 Model 與 View 之間溝通的橋梁，無論 Model 或 View 哪方發生變動，ViewModel 都會即時更新另一方。

Vue.js 所扮演的正是 MVVM 架構中 ViewModel 的角色，也就是「資料狀態」與「畫面顯示」之間溝通的橋梁，如下圖所示，Vue.js 將 DOM 事件監聽程式與資料繫結封裝起來，當 Model 裡面的資料狀態改變時，例如將資料進行運算產生新的結果，Vue.js 會同步更新 View 裡面的畫面顯示；相反的，當 View 裡面的畫面顯示改變時，例如使用者在表單欄位輸入資料或觸發某些事件，Vue.js 會同步更新 Model 裡面的資料狀態，而該資料狀態是由 JavaScript 物件所表示。



參考資料來源：https://012.vuejs.org/guide/#Concepts_Overview