

JavaScript：添加互動性

JavaScript 是一種為網站增添互動性的程式語言。您可以使用它來控制幾乎所有功能——表單資料驗證、按鈕功能、遊戲邏輯、動態樣式、動畫更新等等。本文將引導您入門 JavaScript，並指導您為您的第一個網站添加一些有趣的功能。

什麼是 JavaScript?

JavaScript 是一種成熟的程式語言——它包含您可能在其他程式語言中看到（或至少聽說過）的所有經典程式設計功能，例如變數、循環和函數。

JavaScript 在網頁上使用時（儘管它也可以在其他地方使用）通常透過以下方式運作：

取得一個或多個值（例如數字或頁面上的元素）的引用。

對這些值進行一些操作，例如將數字相加。

傳回結果，以便稍後執行其他操作。例如，您可能希望在頁面上顯示這些數字的總和。

我們來看一個例子。我們將使用前幾篇文章中提到的基本清單：

我們來使用前幾篇可中提到的基本清單：

```
<p>Instructions for life:</p>
<ul>
  <li>Eat</li>
  <li>Sleep</li>
  <li>Repeat</li>
</ul>
```

我們還將定義一個名為.done的 CSS 類，它將為應用該類的任何元素設定樣式，使其看起來像一個已完成的任務，帶有綠色文字和刪除線。下一步，我們將使用 JavaScript 將其應用於元素。

```
.done {
  color: darkseagreen;
  text-decoration: line-through solid black 2px;
}
```


JavaScript 程式碼範例：

現在開始 JavaScript 程式碼。首先，我們把 `` 元素的引用儲存在一個名為 `listItems` 的變數中。然後，定義一個名為 `toggleDone()` 的函數，如果列表項中沒有該類，則添加該類 `done`；如果列表項中已經有該類，則刪除該類。最後，我們循環遍歷列表項目（使用 `forEach()`），並為每個列表項添加一個事件監聽器（使用 `addEventListener()`），這樣當點擊該列表項時，就會切換該 `done` 類，並應用我們先前定義的 CSS。

```
const listItems = document.querySelectorAll("li");
function toggleDone(e) {
  if (!e.target.className) {
    e.target.className = "done";
  } else {
    e.target.className = "";
  }
}
listItems.forEach((item) => {
  item.addEventListener("click", toggleDone);
});
```

應用JavaScript（試用看看）

如果你現在還不理解上面的 JavaScript，也不用擔心。掌握 JavaScript 比掌握 HTML 和 CSS 更具挑戰性，但隨著課程的推進，你會逐漸理解其中的奧秘。

此範例將在 Web 瀏覽器中呈現如下內容：

<https://developer.mozilla.org/en-US/play>

試著點擊清單項目幾次，觀察「完成」樣式是如何切換的。對於這短短的 JavaScript 程式碼來說，效果還不錯。

「Hello world! 」 演練

為了幫助您編寫一些 JavaScript 程式碼，我們將指導您如何將一個Hello world!範例程式碼新增你的範例網站。
(Hello world!是標準的入門程式範例。)

1. 在 first-website 的資料夾中，建立一個名為 scripts 的新資料夾。
2. 在該 scripts 資料夾中，建立一個名為 main.js 的新文字文檔，並儲存它。
3. 轉到您的 index.html 檔案並在結束標記 `</head>` 之前的新行中輸入此程式碼
`<script async src="scripts/main.js"></script>`
4. 它與 CSS 元素的作用相同 `<link>` - 它將 JavaScript 應用於頁面，以便它可以影響 HTML。
5. 將此程式碼新增至您的 scripts/main.js 文件：

```
// Store a reference to the <h1> in a variable
const myHeading = document.querySelector("h1");
// Update the text content of the <h1>
myHeading.textContent = "Hello world!";
```


「Hello world! 」演練

1. 確保 HTML 和 JavaScript 檔案已儲存，然後在瀏覽器中載入 index.html。你應該會看到類似圖示的內容：
2. 讓我們來分析一下這個例子是如何運作的。

我們使用 JavaScript 將標題文字更改為 Hello world!。我們取得了對標題的引用，並將其儲存在名為 myHeading（一個儲存值的容器）的變數中。這類似於將 CSS 應用於元素的方式——首先使用 CSS 選擇器選擇要影響的元素，然後為這些元素定義所需的樣式。在這兩種情況下，當您想要對元素執行某些操作時，都需要先選擇它。

Hello world!

My top level heading

My subheading

My sub-subheading



這是一個單一的段落

At Church, we're a global community of

「Hello world! 」 演練

1. 接下來，我們將 `myHeading` 變數的 `textContent` 屬性值（代表 `<h1>` 元素的文字內容）設定為 *Hello world!*。

以開頭的行 `//` 是 JavaScript 註解。與 HTML 和 CSS 註釋一樣，瀏覽器會忽略這些註釋，這為您提供了一種添加程式碼註釋的方式，以幫助解釋程式碼的工作原理：

讓我們繼續並在我們的範例網站上添加一些新功能。

新增影像變換器

在本節中，你將使用 JavaScript 和 DOM API 功能在兩張圖片之間交替顯示。當使用者點擊顯示的圖片時，就會發生此變化。

1. 選擇另一張圖片作為範例網站的展示圖片。理想情況下，該圖片的尺寸應與您之前添加的圖片相同或盡可能接近。將此圖像保存在您的 `images` 資料夾中。
2. 將以下 JavaScript 程式碼新增至您的 `main.js` 檔案中，確保分別以第一個映像名稱 `a1.jpg` 和第二個 `a2.jpg` 兩個實例。

```
const myImage = document.querySelector("img");
myImage.addEventListener("click", () => {
  const mySrc = myImage.getAttribute("src");
  if (mySrc === "images/a1.jpg") {
    myImage.setAttribute("src", "images/a2.jpg");
  } else {
    myImage.setAttribute("src", "images/a1.jpg");
  }
});
```

儲存所有檔案並 `index.html` 在瀏覽器中載入。現在，當你點擊圖片時，它應該會變成另一張。

在這段程式碼中，你將元素的引用儲存 `` 在 `myImage` 變數中。然後，你給它分配了一個 `click` 事件處理函數。每次 `` 點擊時，函數都會執行以下操作：

- 檢查影像 `src` 屬性的值。
- 使用條件（`if...else` 結構）檢查 `src` 值是否等於原始影像(`a1.jpg`)：
 - 如果是，程式碼會將 `src` 值變更為第二張影像(`a2.jpg`)的路徑，從而強制將另一個影像載入到 `` 元素內。
 - 如果不是（表示影像已更改"`a2.jpg`"），`src` 則值將交換回原始影像(`a1.jpg`)路徑。

注意：本節介紹幾個重要術語。關鍵概念包括

API：一組允許開發人員與程式設計環境互動的功能。Web API（例如我們上面使用的 DOM API 功能）建構在 JavaScript 語言之上，可讓您操作瀏覽器的各個部分及其顯示的網頁。

事件：瀏覽器中發生的事件。它們是實現網站互動的關鍵。您可以使用事件處理函數來執行程式碼來回應事件—這些程式碼區塊在事件發生時運行。最常見的例子是click事件，當使用者點擊某個元素時，瀏覽器會觸發該事件。

函數：一種打包程式碼以便重複使用的方式。你可以在函數中定義一次程式碼，然後根據需要多次運行，這樣可以避免重複編寫相同的程式碼。在本例中，我們定義了一個click事件處理函數，該函數會在使用者每次點擊圖片時執行。

條件語句：用於測試表達式是否回傳true或傳回結果的程式碼結構false，並根據每個結果執行不同的程式碼。if...else語句是條件語句的一種常見形式。

添加個人化歡迎訊息：

接下來，讓我們修改頁面標題，使其在使用者首次造訪網站時顯示個人化的歡迎訊息。

1. 在 `index.html` 中，在結束標記之 `</body>` 前新增以下行：

html複製到剪貼簿

```
<button>Change user</button>
```

2. 在 `main.js` 中，將以下程式碼完全按照原樣放置在文件底部。這將創建對新按鈕和標題的引用，並將每個引用儲存在變數中：

```
let myButton = document.querySelector("button");  
let myHeading = document.querySelector("h1");
```

添加個人化歡迎訊息：

1. 新增以下函數來設定個人化問候語。該函數目前還不會執行任何操作；我們稍後會呼叫它。

```
function setUsername() {  
  const myName = prompt("Please enter your name.");  
  localStorage.setItem("name", myName);  
  myHeading.textContent = `My church is cool, ${myName}`;  
}
```

該 `setUsername()` 函數包含一個 `prompt()` 函數，該函數要求使用者輸入數據，並在使用者按一下「確定」後將其儲存在變數中。在此範例中，我們要求使用者輸入名稱並將其儲存在 `myName`。

2. 接下來，程式碼使用了 Web Storage API，它允許我們將資料儲存在瀏覽器中，並在稍後檢索。我們使用該 `localStorage.setItem()` 函數建立並儲存一個名為的資料項 `"name"`，並將其值設定為 `myName` 包含使用者輸入的變數。

最後，我們將標題的內容 `textContent` 設定為包含使用者儲存名稱的字串。

添加個人化歡迎訊息：

1. 在函數聲明後新增以下條件區塊。這是我們的*初始化程式碼*——它在頁面首次加載時運行，以啟動程式：

```
if (!localStorage.getItem("name")) {  
  setUsername();  
} else {  
  const storedName = localStorage.getItem("name");  
  myHeading.textContent = `My Church is cool, ${storedName}`;  
}
```

該程式碼區塊的第一行使用否定運算子（邏輯非，以!字元表示）來檢查name資料項是否尚未儲存在localStorage。如果不存在，setUsername()則運行函數會建立它。如果存在（即用戶在上次訪問時設定了用戶名），我們使用檢索儲存的名稱localStorage.getItem()，並將標題的設為textContent字串，加上用戶名——就像我們在中所做的那樣setUsername()。

添加個人化歡迎訊息：

1. 為按鈕新增一個事件處理函數。點擊 click 時運行 setUsername()。這允許用戶根據需要儲存不同的名稱。

```
myButton.addEventListener("click", () => {  
  setUsername();  
});
```

2. 保存所有文件並載入 index.html 瀏覽器。系統會立即要求您輸入姓名。輸入後，姓名將 <h1> 作為個人化問候語的一部分顯示在頁面中。請注意，即使重新載入頁面，個人化設定仍然有效。您可以點擊「更改使用者」按鈕輸入新姓名。

使用者名稱是否為空？：

1. 執行範例並出現提示您輸入姓名的對話方塊時，請嘗試按下「取消」按鈕。最終應該顯示一個標題，內容為 *"My Church is cool, null"*。這是因為取消提示時，值被設定為 `null`。在 JavaScript 中，`null` 是一個特殊值，表示不存在值。另外，嘗試按一下「確定」，而不輸入名稱。最終應該會顯示 *"My Church is cool"* 的標題，因為你已將其設為 `myName` 空字串。為了避免這些問題，可以新增另一個條件來檢查使用者是否輸入了空白名稱。將您的 `setUserName()` 函數更新為以下內容：

```
function setUserName() {  
  const myName = prompt("Please enter your name.");  
  if (!myName) {  
    setUserName();  
  } else {  
    localStorage.setItem("name", myName);  
    myHeading.textContent = `My Church is cool, ${myName}`;  
  }  
}
```

使用者名稱是否為空？：

1. 用人類語言來說，這意味著：如果它 `myName` 沒有值，`setUserName()` 則從頭開始運行。如果它有值（如果上述語句不成立），則將該值儲存進去 `localStorage`，並將其設為標題的文字。

結論

如果您已遵循本文中的所有說明，最終應該會看到類似附圖的頁面：



My Church is cool, JOSEPH

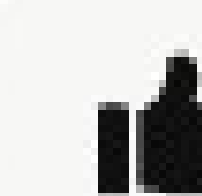


這是一個單一的段落

At Church, we're a global community of



發佈你的網站



code參考這裡

<https://github.com/mintetang/mintang.github.io/blob/main/js/main.js>