

## CHAPTER 07

# DOM與JAVASCRIPT

HORAZON

互動媒體設計

# 什麼是 JAVASCRIPT?

JAVASCRIPT (簡稱 JS)

它**是**一種程式語言 (PROGRAMMING LANGUAGE)

負責網頁的**互動**、**邏輯**與**資料處理** (大腦與肌肉)

[!WARNING]

JavaScript 與 Java 是完全不同的兩種語言！就像「熱狗」與「狗」的關係。

- HTML (結構/骨架)：網頁有什麼內容？(標題、圖片、按鈕)
- CSS (樣式/皮膚)：網頁長什麼樣子？(顏色、字體、排版)
- JavaScript (行為/動作)：網頁能做什麼？(點擊、運算、變更內容)

JavaScript 讓靜態的網頁動起來，產生**互動性**。

# HTML 內的 JAVASCRIPT

JavaScript 程式碼必須放在 `<script>` 標籤內：

```
<!DOCTYPE html>
<html>
<body>

  <h1>我的第一個 JS</h1>

  <script>
    alert('哈囉，JS！');
  </script>

</body>
</html>
```

通常建議將 `<script>` 放在 `<body>` 的**最下方**，確保網頁元素載入完成後再執行程式。

# 輸出方式 (OUTPUT)

如何讓程式「說話」或顯示結果？

- `alert()`：跳出警告視窗 (最簡單，但會中斷操作)。
- `console.log()`：在瀏覽器控制台 (F12) 顯示訊息 (開發者常用)。
- `document.querySelector().innerText`：直接改變網頁內容。

```
alert("跳出視窗");  
console.log("這是在控制台顯示的訊息");
```

# 變數 (VARIABLES)

用來儲存資料的箱子。

現代 JS 推薦使用 `let` 與 `const`：

- `let`：宣告**可變動**的變數 (例如：分數、計數器)。
- `const`：宣告**常數** (不可變動，例如：圓周率、網站網址)。

```
let score = 100;  
score = 95;           // OK，可以改變  
  
const pi = 3.14;  
// pi = 3.14159; // Error! 不能改變 const
```

# 資料型態 (DATA TYPES)

變數可以存放不同類型的資料：

- **字串 (String)**：文字，需用引號包起來 ( "文字" 或 '文字' )。
- **數字 (Number)**：整數或小數 ( 10 , 3.5 )。
- **布林值 (Boolean)**：只有 true (真) 或 false (假)。

```
let name = "Horazon"; // 字串
let age = 18;          // 數字
let isTeacher = true;  // 布林值
```

# 函式 (FUNCTIONS)

將一段程式碼包裝起來，需要時再呼叫使用 (像是「技能」或「SOP」)。

```
// 定義函式
function sayHello() {
    alert("你好!");
}

// 呼叫函式 (執行)
sayHello();
```



# DOM 操作 (DOCUMENT OBJECT MODEL)

JavaScript 透過 DOM 來控制 HTML 元素。

最常用的選取方式：`document.querySelector()`

```
<h1 id="title">原本的標題</h1>
```

```
// 選取 ID 為 title 的元素
let myTitle = document.querySelector("#title");

// 修改文字內容
myTitle.innerText = "被 JS 修改後的標題";

// 修改樣式
myTitle.style.color = "red";
```

# 事件 (EVENTS)

偵測使用者的行為，例如點擊、滑鼠移入、鍵盤輸入。

最常用的是 `onclick` (點擊事件)。

```
<button onclick="changeText()">點我改變標題</button>
<h1 id="demo">你好</h1>

<script>
function changeText() {
  let el = document.querySelector("#demo");
  el.innerText = "文字變了！";
  el.style.color = "blue";
}
</script>
```

# 條件判斷 (IF...ELSE)

如果...就...，否則... (邏輯判斷)。

```
let score = 80;  
  
if (score >= 60) {  
    console.log("及格!");  
} else {  
    console.log("不及格...");  
}
```

# 實作練習：計數器 (COUNTER)

建立一個網頁，包含：

1. 一個數字顯示 `0` (使用 `<h1>` 或 `<span>`)。
2. 一個「+1」按鈕。
3. 一個「重置」按鈕。

**功能要求：**

- 點擊「+1」，數字會增加。
- 點擊「重置」，數字歸零。

# 實作練習：計數器 (參考解答)

```
<h1>目前數字：<span id="count">0</span></h1>
<button onclick="add()">+1</button>
<button onclick="reset()">重置</button>

<script>
  let number = 0; // 變數記住目前的數字
  let el = document.querySelector("#count");

  function add() {
    number = number + 1; // 數字加一
    el.innerText = number; // 更新畫面
  }

  function reset() {
    number = 0; // 歸零
    el.innerText = number; // 更新畫面
  }
</script>
```

# 結論

1. JavaScript 是網頁的「大腦」，負責邏輯與互動。
2. 使用 `<script>` 撰寫 JS 程式碼。
3. 變數 ( `let` / `const` ) 用來存資料。
4. 函式 ( `function` ) 用來定義動作。
5. DOM ( `document.querySelector` ) 讓 JS 能控制 HTML。
6. 事件 ( `onclick` ) 讓網頁能回應使用者的操作。

# 重點回顧

1. **結構**：放在 `<body>` 結束前。
2. **語法**：分號 `;` 結尾 (建議)，區分大小寫。
3. **除錯**：善用 `console.log()` 與 F12 開發者工具。
4. **核心**：選取元素 -> 監聽事件 -> 修改內容/樣式。