

JavaScript 入門指南 (電腦學生適用)

[TOC]- [JavaScript 入門指南 \(電腦學生適用\)](#)

- [1. 前言](#)
 - [1.1. 什麼是 JavaScript ?](#)
 - [1.2. JavaScript 的用途](#)
 - [1.3. 開發環境設定](#)
- [2. JavaScript 基礎](#)
 - [2.1. 語法基礎](#)
 - [2.1.1. 變數](#)
 - [2.1.2. 資料類型](#)
 - [2.1.3. 運算符](#)
 - [2.1.4. 控制流程](#)
 - [2.1.5. 函數](#)
 - [2.1.6. 物件](#)
 - [2.1.7. 陣列](#)
- [3. DOM 操作](#)
 - [3.1. 什麼是 DOM ?](#)
 - [3.2. 選取元素](#)
 - [3.3. 修改元素](#)
 - [3.4. 新增和刪除元素](#)
- [4. 事件處理](#)
 - [4.1. 什麼是事件 ?](#)
 - [4.2. 事件監聽器](#)
 - [4.3. 常見事件](#)
- [5. JavaScript 與 HTML 和 Python 的結合](#)
 - [5.1. HTML、CSS 和 JavaScript](#)
 - [5.2. Python 和 JavaScript](#)
 - [5.3. 前後端通訊](#)
- [6. 進階主題 \(選修\)](#)
 - [6.1. 非同步 JavaScript](#)
 - [6.2. AJAX 和 JSON](#)
 - [6.3. ES6+ 新特性](#)
 - [6.4. JavaScript 框架和函式庫](#)
- [7. 學習資源](#)
- [8. 結語](#)

1. 前言

1.1. 什麼是 JavaScript ？

JavaScript 是一種直譯式、基於原型的程式語言，主要用於使網頁具有互動性。它可以在瀏覽器中執行，操作網頁內容 (HTML 和 CSS)。

1.2. JavaScript 的用途

- **前端開發**：建立互動式使用者介面、處理使用者輸入、動態更新網頁內容。
- **後端開發**：使用 Node.js 建立伺服器端應用程式。
- **行動應用程式開發**：使用 React Native 或 NativeScript 等框架建立跨平台行動應用程式。
- **遊戲開發**：使用 Phaser 或 Three.js 等框架建立網頁遊戲。

1.3. 開發環境設定

- **文字編輯器**：建議使用 Visual Studio Code (VS Code)。
- **瀏覽器**：Chrome、Firefox、Safari 等現代化的網頁瀏覽器。

2. JavaScript 基礎

2.1. 語法基礎

2.1.1. 變數

使用 `var`、`let` 或 `const` 關鍵字宣告變數。

```
let message = "Hello";
message = "World"; // 可以重新賦值

const PI = 3.14159;
// PI = 3.14; // 錯誤！不能重新賦值 const 變數
```

2.1.2. 資料類型

- 原始類型：
 - `String` (字串): 例如 `"Hello"`、`'World'`
 - `Number` (數字): 例如 `10`、`3.14`
 - `Boolean` (布林): `true` 或 `false`
 - `Null` (空值): `null`
 - `Undefined` (未定義): `undefined`
 - `Symbol` (符號): ES6 引入，用於建立唯一識別符。
 - `BigInt` (大整數): 用於表示大於 `Number` 安全範圍的整數。
- 物件類型：
 - `Object` (物件): 用於儲存鍵值對的集合。
 - `Array` (陣列): 用於儲存有序的資料列表。
 - `Function` (函數): 用於封裝可重複使用的程式碼區塊。

2.1.3. 運算符

- 算術運算符：`+` (加)、`-` (減)、`*` (乘)、`/` (除)、`%` (取餘數)
- 賦值運算符：`=` (賦值)、`+=` (加等於)、`-=` (減等於)、`*=` (乘等於)、`/=` (除等於)
- 比較運算符：`==` (等於)、`!=` (不等於)、`>` (大於)、`<` (小於)、`>=` (大於等於)、`<=` (小於等於)、`===` (嚴格等於，包含類型檢查)、`!==` (嚴格不等於)
- 邏輯運算符：`&&` (且)、`||` (或)、`!` (非)

2.1.4. 控制流程

- `if...else` 語句：

```
let age = 20;
if (age >= 18) {
  console.log("成年人");
} else {
```

```
    console.log("未成年人");  
}
```

- **switch** 語句：

```
let day = "Monday";  
switch (day) {  
  case "Monday":  
    console.log("星期一");  
    break;  
  case "Tuesday":  
    console.log("星期二");  
    break;  
  default:  
    console.log("其他日子");  
}
```

- 迴圈：

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

2.1.5. 函數

```
function greet(name) {  
  console.log("Hello, " + name + "!");  
}  
  
greet("Alice"); // 呼叫函數  
  
function add(a, b) {  
  return a + b;  
}
```

```
let sum = add(5, 3); // sum 的值為 8
```

2.1.6. 物件

```
let person = {  
  name: "Bob",  
  age: 30,  
  city: "New York"  
};  
  
console.log(person.name); // 存取屬性  
console.log(person["age"]); // 另一種存取屬性的方式
```

2.1.7. 陣列

```
let numbers = [1, 2, 3, 4, 5];  
  
console.log(numbers[0]); // 存取第一個元素  
numbers.push(6); // 在陣列末尾新增元素``  
  
### 2.2. Hello World  
  
``html  
<!DOCTYPE html>  
<html>  
<head>  
  <title>Hello World</title>  
</head>  
<body>  
  <h1>Hello, World!</h1>  
  <script>  
    alert("Hello, World!");  
  </script>  
</body>  
</html>
```

3. DOM 操作

3.1. 什麼是 DOM ?

DOM (Document Object Model) 是網頁的結構化表示。JavaScript 可以使用 DOM 來存取和操作 HTML 元素。

3.2. 選取元素

- `document.getElementById(id)`: 通過 ID 選取元素。
- `document.getElementsByClassName(className)`: 通過類別名稱選取元素。
- `document.getElementsByTagName(tagName)`: 通過標籤名稱選取元素。
- `document.querySelector(selector)`: 使用 CSS 選擇器選取第一個匹配的元素。
- `document.querySelectorAll(selector)`: 使用 CSS 選擇器選取所有匹配的元素。

3.3. 修改元素

- `element.innerHTML`: 修改元素的 HTML 內容。
- `element.textContent`: 修改元素的文字內容。
- `element.setAttribute(attributeName, attributeValue)`: 設定元素的屬性。
- `element.style.property`: 修改元素的 CSS 樣式。

3.4. 新增和刪除元素

- `document.createElement(tagName)`: 建立一個新的 HTML 元素。
- `element.appendChild(newElement)`: 將新元素新增到元素的子節點列表末尾。
- `element.removeChild(childElement)`: 從元素中刪除子元素。

4. 事件處理

4.1. 什麼是事件？

事件是網頁上發生的事情，例如使用者點擊按鈕、滑鼠移動、鍵盤按下等。

4.2. 事件監聽器

```
let button = document.getElementById("myButton");
button.addEventListener("click", function() {
  alert("按鈕被點擊了！");
});
```

4.3. 常見事件

- `click`: 點擊事件
- `mouseover`: 滑鼠懸停事件
- `mouseout`: 滑鼠離開事件
- `keydown`: 鍵盤按下事件
- `keyup`: 鍵盤釋放事件
- `submit`: 表單提交事件
- `load`: 頁面載入完成事件

5. JavaScript 與 HTML 和 Python 的結合

5.1. HTML、CSS 和 JavaScript

HTML 負責網頁的結構和內容，CSS 負責網頁的樣式，JavaScript 負責網頁的互動性和動態性。

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Example</title>
</head>
<body>
  <button id="myButton">Click Me</button>
  <p id="myParagraph">This is a paragraph.</p>

  <script>
    // 獲取按鈕和段落元素
    let button = document.getElementById("myButton");
    let paragraph = document.getElementById("myParagraph");

    // 當按鈕被點擊時，改變段落的文字
    button.addEventListener("click", function() {
      paragraph.textContent = "Button was clicked!";
    });
  </script>
</body>
</html>
```

5.2. Python 和 JavaScript

Python 通常用於建立網頁的後端，JavaScript 通常用於建立網頁的前端。

5.3. 前後端通訊

前端和後端通過 API 進行通訊，常見的 API 格式包括 JSON。

- 前端 (JavaScript):

```
fetch('/api/data') // 向後端發送請求
  .then(response => response.json()) // 將響應轉換為 JSON
  .then(data => {
    // 使用資料更新網頁
    document.getElementById('dataContainer').textContent = data.message;
  });
```

- 後端 (Python - 使用 Flask):


```
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/api/data')
def get_data():
    data = {'message': 'Hello from the Python backend!'}
    return jsonify(data) # 回傳 JSON 格式的資料

if __name__ == '__main__':
    app.run(debug=True)
```

6. 進階主題 (選修)

6.1. 非同步 JavaScript

- 回調函數 (Callbacks)
- Promise
- Async/Await

6.2. AJAX 和 JSON

AJAX (Asynchronous JavaScript and XML) 可以在不重新載入整個頁面的情況下，與伺服器交換資料。JSON (JavaScript Object Notation) 是一種輕量級的資料交換格式。

6.3. ES6+ 新特性

- 箭頭函數 (Arrow Functions)
- 類別 (Classes)
- 模組 (Modules)
- 解構賦值 (Destructuring)
- 展開運算符 (Spread Operator)
- 其餘參數 (Rest Parameters)

6.4. JavaScript 框架和函式庫

- React
- Angular
- Vue.js
- jQuery

7. 學習資源

- **MDN Web Docs:** <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript>
- **freeCodeCamp:** <https://www.freecodecamp.org/>
- **Codecademy:** <https://www.codecademy.com/>

8. 結語

希望這份指南能幫助你入門 JavaScript！ 祝你學習愉快！

