

【JS】 ?? 與 || 比較

在 JavaScript 中，`??` (Nullish Coalescing) 和 `||` (Logical OR) 看起來很像，都是用來提供預設值的運算子，但它們判斷「什麼時候該用預設值」的標準完全不同。

這篇教學將幫你釐清兩者的核心差異，並告訴你什麼時候該用哪一個。

核心差異：判斷標準

兩者的主要區別在於對於「虛值 (Falsy values)」的處理方式。

1. 邏輯或運算子 `||` (Logical OR)

只要左側的值轉換為布林值後是 `false`，就會回傳右側的值。

JavaScript 中的 Falsy 值包括：

- `null`
- `undefined`
- `0`
- `""` (空字串)
- `NaN`
- `false`

2. 空值合併運算子 `??` (Nullish Coalescing)

只有當左側的值是 `null` 或 `undefined` 時，才會回傳右側的值。

它會保留 `0`、`""` 和 `false` 等有效數據。

對比表

| 特性 | <code> </code> (OR) | <code>??</code> (Nullish) |
|-----------------------------|--------------------------------|---|
| 觸發預設值的條件 | 任何 Falsy 值 (0, "", false...) | 僅限 <code>null</code> 或 <code>undefined</code> |
| 對 <code>0</code> 的處理 | 視為 <code>false</code> ，會被預設值覆蓋 | 視為有效值，保留 <code>0</code> |
| 對 <code>""</code> (空字串) 的處理 | 視為 <code>false</code> ，會被預設值覆蓋 | 視為有效值，保留 <code>""</code> |
| 設計目的 | 廣泛的邏輯備案 | 精確處理「缺失資料」 |

實戰範例

假設我們正在開發一個使用者設定頁面：

JavaScript

```
const userSettings = {
  speed: 0,
  nickname: "",
  isPremium: false,
  theme: null
};

// --- 使用 || 的結果 ---
console.log(userSettings.speed || 10); // 輸出: 10 (即便 0 是有效的數值)
console.log(userSettings.nickname || "匿名"); // 輸出: "匿名" (即便空字串是有效的名字)
```

```
console.log(userSettings.isPremium || true); // 輸出: true (這會讓使用者的 false 設定失效)
```

```
// --- 使用 ?? 的結果 ---
```

```
console.log(userSettings.speed ?? 10); // 輸出: 0 (正確保留了數值)  
console.log(userSettings.nickname ?? "匿名"); // 輸出: "" (正確保留了空字串)  
console.log(userSettings.isPremium ?? true); // 輸出: false (正確保留了布林值)  
console.log(userSettings.theme ?? "Light"); // 輸出: "Light" (因為是 null)
```

使用場景建議

什麼時候使用 `??` ? (推薦做法)

當你想要**「安全地提供預設值」**且不希望意外覆蓋掉 `0`、`false` 或 `""` 時。

- 處理 API 回傳的選擇性參數。
- 數值計算 (例如: 計數器、座標)。
- UI 表單輸入 (允許空字串)。

什麼時候使用 `||` ?

當你想要**「只要沒內容就給預設值」**，或者你確定資料只會是字串/物件，且你不需要處理 `0` 或空字串的情況。

- 舊版瀏覽器相容性 (`??` 是 ES2020 引入的)。
- 確保變數一定有「實質內容」(非空、非零)。

注意事項：混合使用

在 JavaScript 中，出於運算優先級的安全性考慮，你**不能**在沒有括號的情況下直接混合使用 `&&`、`||` 與 `??`：

JavaScript

```
// ❌ 錯誤語法: SyntaxError  
// const result = a || b ?? c;  
  
// ✅ 正確語法: 使用括號明確優先級  
const result = (a || b) ?? c;
```

“**總結建議**：在現代開發中，為了避免處理數值 `0` 或布林值時出現非預期的 Bug，**優先考慮使用 `??`**。”

需要我為你示範如何將現有的舊代碼從 `||` 重構成 `??` 嗎？

🕒 修訂版本 #3

★ 由 treeman 建立於 29 🕒🕒🕒🕒 2025 09:25:08

✍ 由 treeman 更新於 29 🕒🕒🕒🕒 2025 09:36:37