

# 【ES6】null 判斷

[https://developer.mozilla.org/zh-TW/docs/Web/JavaScript/Reference/Operators/Optional\\_chaining](https://developer.mozilla.org/zh-TW/docs/Web/JavaScript/Reference/Operators/Optional_chaining)

**可選串連運算子** `?.` 允許進行深層次的物件值存取，而無需透過明確的物件值串連驗證。`?.` 運算子的操作與 `.` 屬性存取運算子相似，後者會在參照到 `nullish` (`null` or `undefined`) 的值時出現錯誤，而前者可選串連則回傳 `undefined`。當需要存取一個函數，而這函數並不存在時，則會回傳 `undefined`

```
const adventurer = {
  name: 'Alice',
  cat: {
    name: 'Dinah',
  },
};

const dogName = adventurer.dog?.name;
console.log(dogName);
// Expected output: undefined

console.log(adventurer.someNonExistentMethod?.());
// Expected output: undefined
```

當一個物件 `obj` 是巢狀結構時，在沒有可選串連之下，要去查找一深層的屬性值需要驗證每層間的參照連結：

```
let nestedProp = obj.first && obj.first.second && obj.first.second.third;
```

現在可以用 `?.` 做串接判斷

```
let nestedProp = obj.first?.second?.third
```

在用 `??` 賦予預設值避免 `undefined`

```
let nestedProp = obj.first?.second?.third ?? "default"
```

🕒 修訂版本 #1

★ 由 treeman 建立於 10 🕒 2024 11:44:57

✍ 由 treeman 更新於 10 🕒 2024 11:52:06