



JavaScript 程式設計新手村

單元15 - JavaScript 物件 Object 基礎（上）

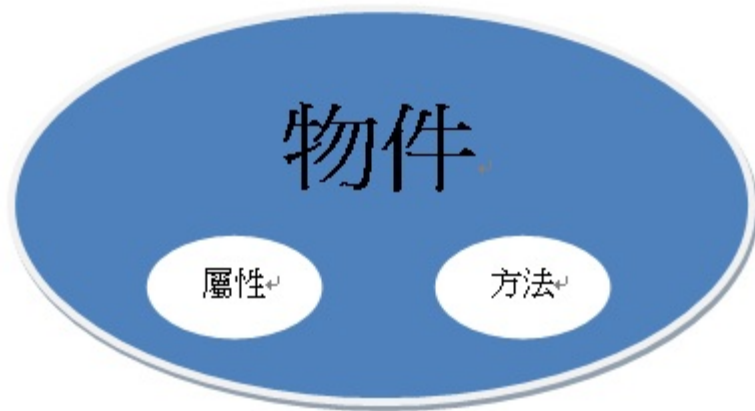
@kdchang

Outline

1. 物件基礎概念
2. JavaScript 常用內建物件

物件基礎概念

Object 能夠理解為現實生活中真實的物件



在 JavaScript 中，物件是有屬性及方法獨立的實體，例如：生活中的物件像是貓，貓物件有顏色、名字、年紀，有喵喵叫的方法

JavaScript 支援的物件

1. 內建物件

JavaScript 語言中原生的物件，一般有11種（String、Array、Boolean、Date、Global、Math、Number、Object、RegExp、Error），其中 `Object` 為所有物件老祖宗

2. 自訂物件

由使用者自己定義的物件

3. 宿主物件（Host Objects）

由宿主提供物件（如：瀏覽器），像是 BOM（Browser Object Model）

JavaScript 常用內建物件

JavaScript 常用內建物件

在 JavaScript 中共有 11 種內建物件，其中 String、Array、Date、Math、Error 和 RegExp 物件較常使用特別說明

字串 **String** 物件

字串的使用有兩種方式，除了基本資料型別外，也可以使用 `new` 物件的方式建立，兩者都可以使用字串物件的屬性和方法，唯基本資料型別無法使用 `prototype` 屬性

```
const str1 = 'JS';  
const str2 = new String('HI JS');
```


字串 **String** 物件方法

字串長度和大小寫

`length` 屬性可以取得字串長度

`toLowerCase()` 方法可以將英文大寫轉小寫

`toUpperCase()` 方法可以將英文小寫轉大寫

字串 **String** 物件方法

取得字串指定字元

- `charAt(index)` : 取得 `index` 位置的字元
- `charCodeAt(index)` : 取得 `index` 位置的 **Unicode**

字串搜尋

- `indexOf(string, index)` : 傳回第一次搜尋到字串的`index`，未找到傳回-1，傳入字串為欲搜尋字串，`index` 為開始位置
- `lastIndexOf(string)` : 同上一方法，不過是從尾部反向搜尋
- `match(string)` : 同第一種方式，但傳回找到字串，無則傳`null`
- `search(string)` : 與 `indexOf` 功能類似，但可用正則式

範例程式

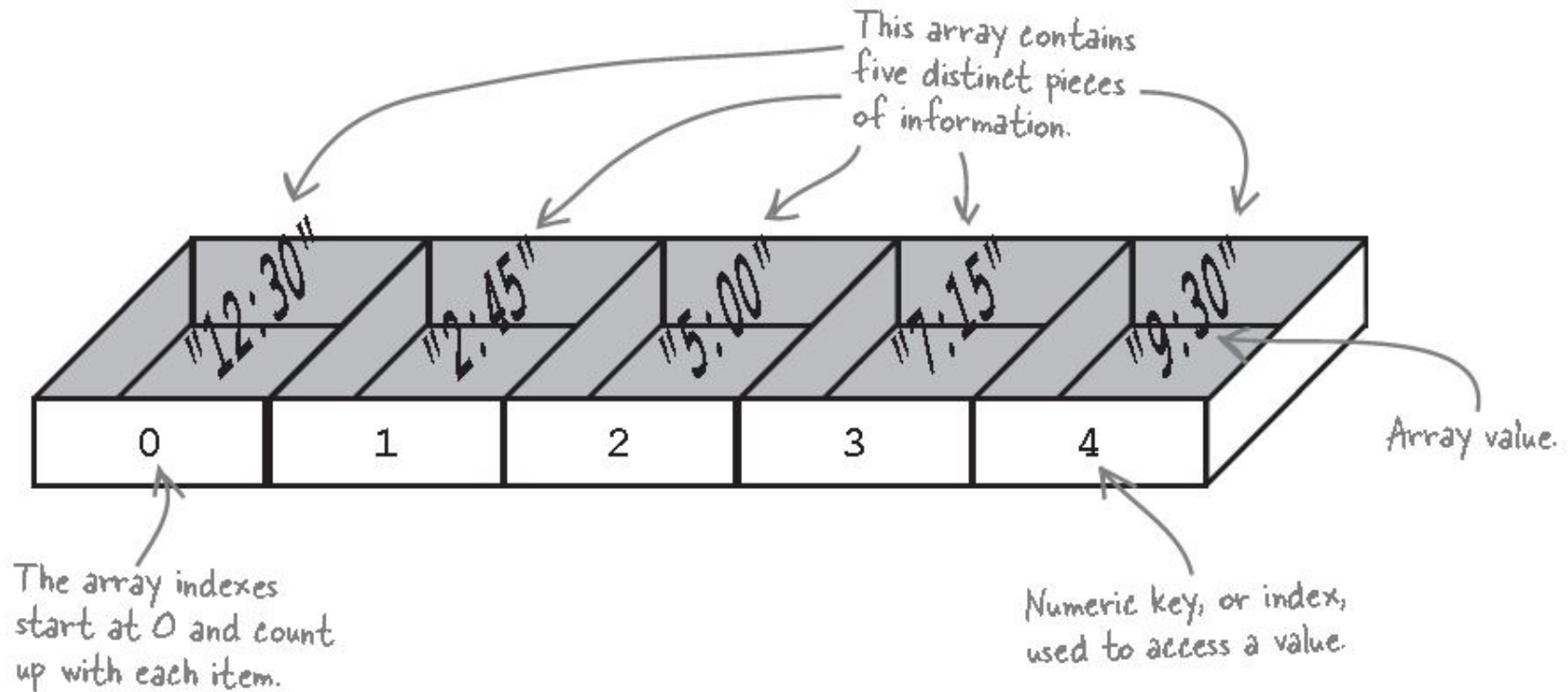
字串 **String** 物件方法

子字串的處理

- `replace(string1, string2)` : 將找到的 `string1` 子字串取代為 `string2`
- `split(string)` : 使用參數 `string` 作為分割，並回傳分割後 `Array` 物件
- `substr(index, length)` : 從 `index` 開始取出`length`個字元
- `substring(index1, index2)` : 取 `index1` 到 `index2`(不含)子字串
- `concat(string)` : 將 `string` 新增到 `string` 物件字串之後

範例程式

陣列 array 物件



- array（陣列）為程式語言常使用的儲存用資料結構
- 陣列中元素型別可不同，能動態新增元素（其他語言不行）
- JavaScript 沒有物件型別，每個 array 元素是 Array 物件的屬性
- 陣列元素 index 從 0 開始

產生陣列 `array` 物件

使用 `new Array` 產生陣列：

```
// 宣告有三個元素的陣列  
const nameA = new Array(3);  
nameA[0] = 'KD';  
nameA[1] = 'XD';  
nameA[2] = 'XDD';  
  
// 也可以在建立物件時建立元素值  
const num = new Array(100, 22, 34);
```

更簡單，使用 `[]`：

```
const nameB = [];  
nameB[0] = 'KD';  
nameB[1] = 'XD';  
nameB[2] = 'XDD';
```

Array 物件內建屬性和方法

- `length` 屬性：取得陣列長度
- `join()`：將陣列元素用 "逗號," 串成一個字串
- `reverse()`：將陣列元素反轉
- `sort()`：將陣列元素依照 `unicode` 進行排序
- `concat(array)`：將參數中的陣列合併到目前陣列

範例程式

走訪一維陣列(印出所有元素)

```
for(var i = 0; i < 3; i++){  
    //建議條件不要寫死，3 可以改使用 num.Length  
    document.write(num[i]);  
}
```

二維 Array 物件

在 JavaScript 中並不能直接建立二維或多維陣列，所以需要透過於陣列元素中再建立陣列物件來建立多維陣列

```
const users = new Array(2);

for(var i = 0; i < 2; i++){
    users[i] = new Array(2);
}
```


Date 物件

透過 Date 物件取得電腦日期和時間

```
const d = new Date();  
// milliseconds since 1970/01/01  
const now = Date.now();  
const date = d.getDate();  
const year = d.getFullYear();  
  
console.log(now); // 1970/1/1到現在的毫秒數  
console.log(date); // 日期  
console.log(year); // 年
```

Math 物件

Math 物件中擁有數學常數和函數的屬性和方法，Math 物件和 Global 物件一樣為腳本語言引擎內建，不需要 new 運算子自己就會建立，故可以直接使用屬性和方法

```
console.log(Math.PI);  
console.log(Math.E);  
console.log(Math.LN2);
```

Math 物件方法

Math物件的亂數最大和最小值

- `max(value1, value2)` : 傳回兩個參數中的最大值
- `min(value1, value2)` : 傳回兩個參數中的最小值
- `random()` : 傳回亂數值 (0-1, 不含1)
- `round(value)` : 將參數四捨五入後傳回

Math物件的數學方法

- `abs()` : 傳回絕對值
- `log()` : 傳會自然對數
- `sqrt()` : 傳回平方根
- `pow()` : 傳回次方
- `exp()` : 自然對數

Error 物件

在寫程式時常常會有不預期的錯誤，這時候就需要有錯誤處理的機制來幫我們做處理，否則程式出錯後可能會當機或是發生不可預期的錯誤，在 JavaScript 中 Error 物件會儲存執行時發生錯誤的資訊，發生錯誤時會自動建立

基本錯誤處理架構

```
try {  
    // 此區塊內為需要例外處理的程式碼  
}  
catch(e) {  
    // 如果try區塊發生錯誤會執行此一區塊，並傳入e參數(Error 物件)  
}  
finally {  
    // 這是選擇性區域，是不管錯誤是否發生都會執行的區塊  
}
```

RegExp 物件

RegExp 是 JavaScript 用來使用正規表示法的物件，用以比對其它字串是法符合範本字串，以便進一步對字串進行操作，功能強大

- `exec(string)`：檢查參數string字串是否符合正規運算是子字串，符合回傳字串陣列
- `test(string)`：檢查參數string字串是否符合正規運算是子字串，符合回傳true不符合回傳false

```
// \S+ 至少有一非空白字元
const matches = /(hello \S+)/.exec('This is a hello world!');
console.log(matches[1]); // hello world!

const str = "hello world!";
const result = /^hello/.test(str);
console.log(result); // true
```

總結

在這個章節中我們學會了：

1. 物件基礎概念
2. JavaScript 常用內建物件