

ES6七大新特性

解構賦值





解構賦值(Destructuring Assignment)語法

解構賦值是”**解析結構+指定值運算**”的意思。是專門設計給**物件**與**陣列**使用的指定值語法，以”**鏡子**”般的對映樣式，提取物件與陣列中的成員值。設計此語法的目的是讓程式碼更簡短與提高閱讀性。

```
const [a, b] = [1, 2]
const {a: x, b: y} = {a: 1, b: 2}
const {a, b} = {a: 1, b: 2}
```

註：可使用指定預設值、可搭配函式傳入參數與其餘運算符使用



- ✓ 總是使用const宣告來作解構賦值
- ✓ 解構賦值的樣式中不要包含空樣式(空物件或空陣列)
- ✓ 在函式的傳入參數或回傳值中作解構賦值時，優先使用物件

ES6七大新特性

展開與其餘運算符





展開與其餘運算符(Spread Operator & Rest Operator)語法

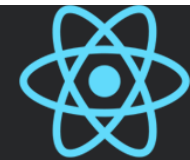
展開運算符 - 展開一個陣列為一個個的獨立值，用於”陣列字面”與”函式呼叫”

```
const c = [...arr, b]  
f(...arr)
```

其餘運算符 - 集合所有剩餘的值，組合成一個陣列。用於”函式傳入參數定義”與”解構賦值”

```
function f(...a) { }  
const [a, ...b] = [1, 2, 3]
```

註：其餘參數在傳入參數定義中，必定是位於最後一位，並且在參數中只能有一個其餘參數。



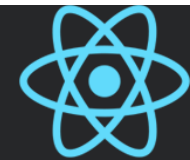
- ✓ 不要使用函式中的arguments物件，總是使用其餘參數語法來取代它
- ✓ 不要在展開運算符與其餘運算符後面有空格
- ✓ 用展開運算符來作拷貝陣列，取代函式的`apply`與陣列的`concat`的語法

ES6七大新特性

類別



類別(Class)語法



原型物件導向的語法糖，轉換ES5語法後為函式。目的是提供一種用於自訂類型、物件的繼承與擴充時，更簡便的語法。

```
class MyClass extends Parent{  
  constructor( ) {  
    super( )  
  }  
}
```

註：類別是React用於撰寫元件的主要語法，注意其中有部份是超出ES6標準(ES7+)的語法。



- ✓ 在命名類別時，使用大駝峰(PascalCase)命名方式
- ✓ 撰寫自訂的toString()方法時要確保它是可以運作的，而且不會產生副作用
- ✓ 不要使用JavaScript中的getters/setters，可能有不預期的副作用，不易測試與維護

ES6七大新特性

模組系統



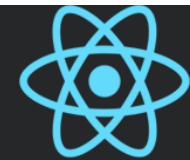
模組(Module)語法



組織與管理程式碼檔案的重要語法，搭配npm工具，方便使用與導入現有模組。

```
// 模組輸出的檔案  
export default MyClass  
  
// 模組導入的檔案  
import MyClass from './MyClass'
```

註：最新的瀏覽器內建支援情況都尚未完整，要先用如babel工具進行編譯/打包才能使用。



- ✓ 不要使用萬用符(wildcard, *)作導入模組
- ✓ 從一個位置只作一次導入
- ✓ 導入語句都放在程式碼檔案中其他語句的上面
- ✓ 如果在模組中只需要單一個輸出，優先使用預設(default)輸出的語法