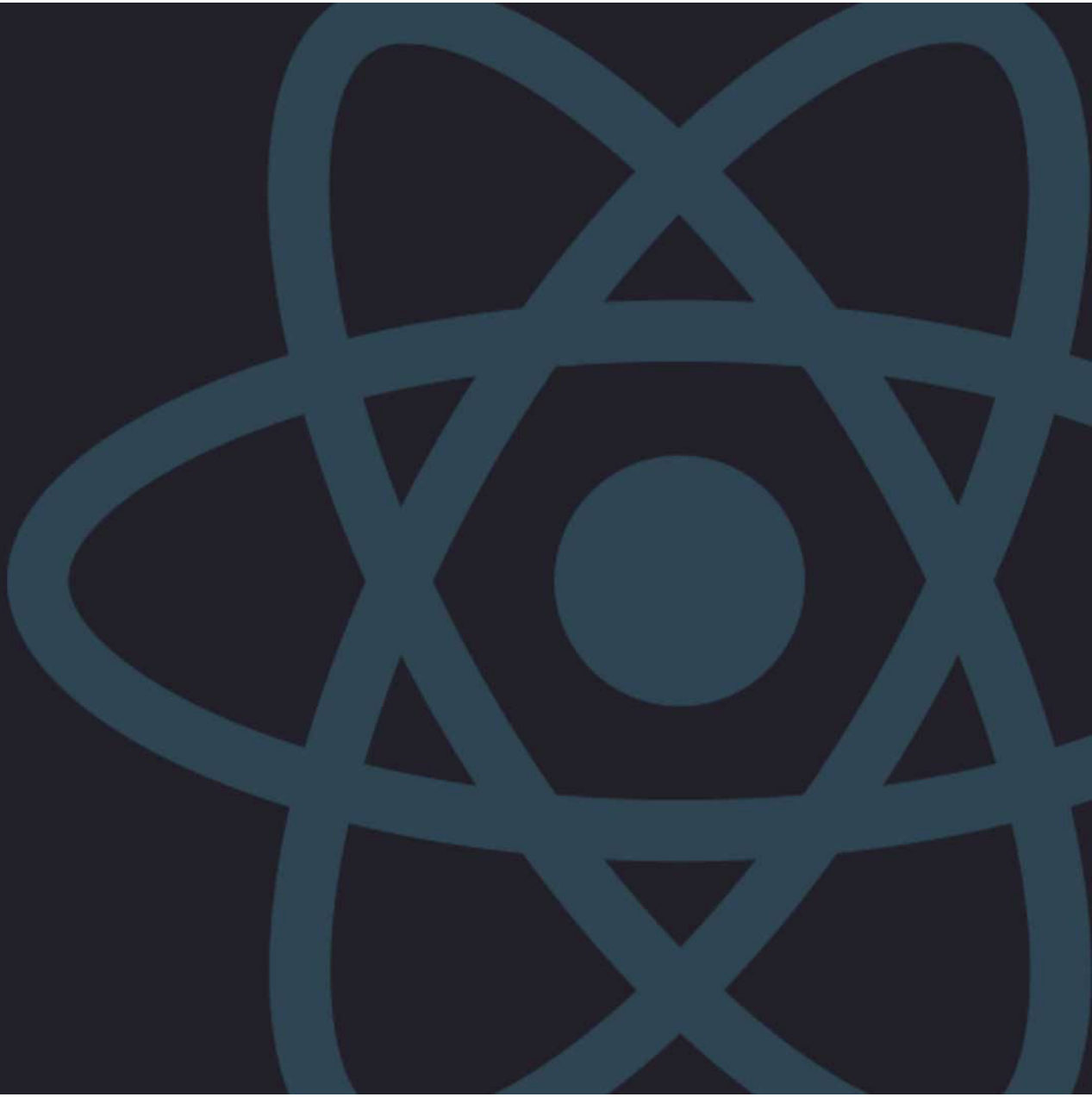
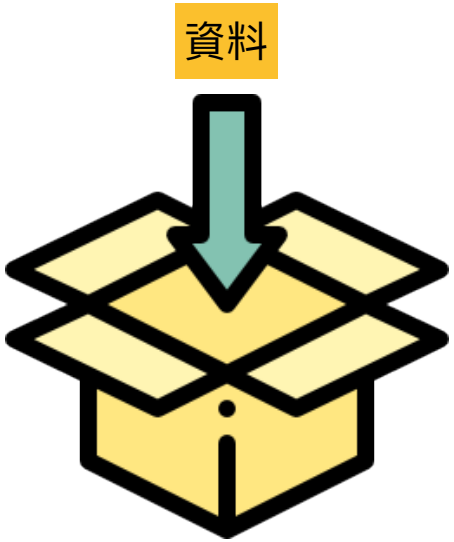
JavaScript(ES6)入門

常數&變數



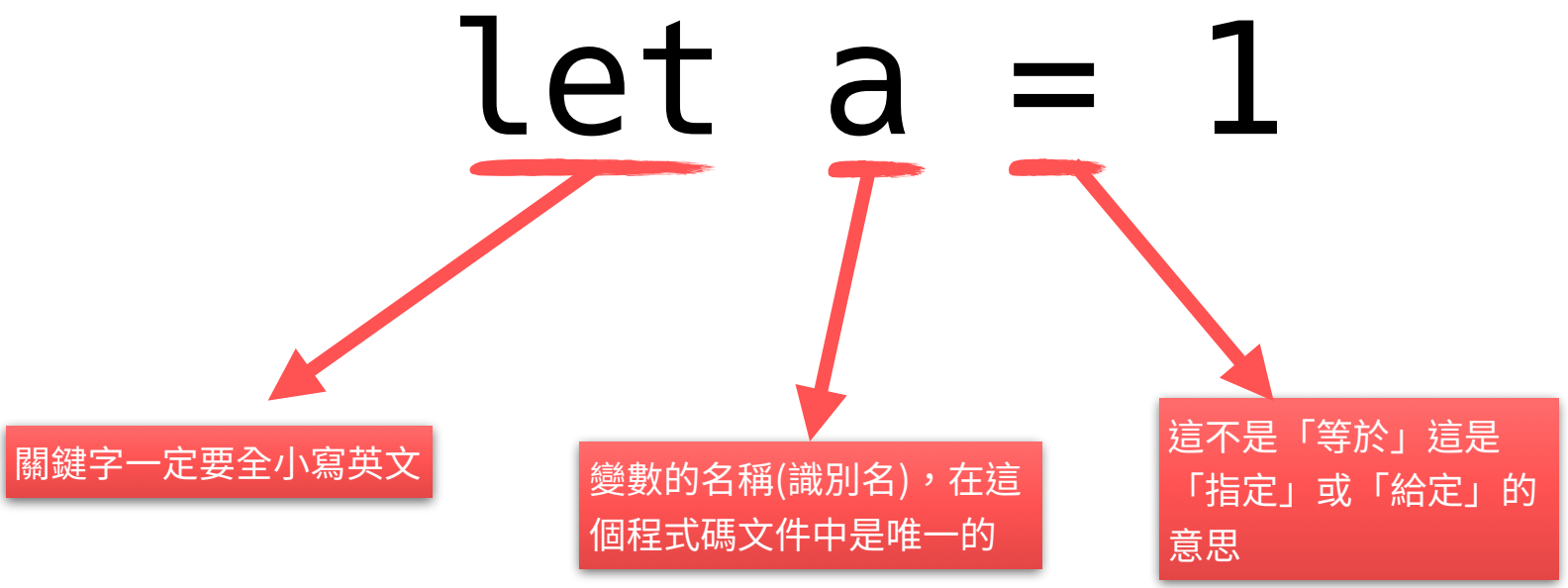
變數

變數 - 有「盒子」的概念，把資料放在裡面，然後標註上一個名稱作為標籤



變數

宣告時使用 **let** 關鍵字，一開始宣告時要養成給定初始值的好習慣



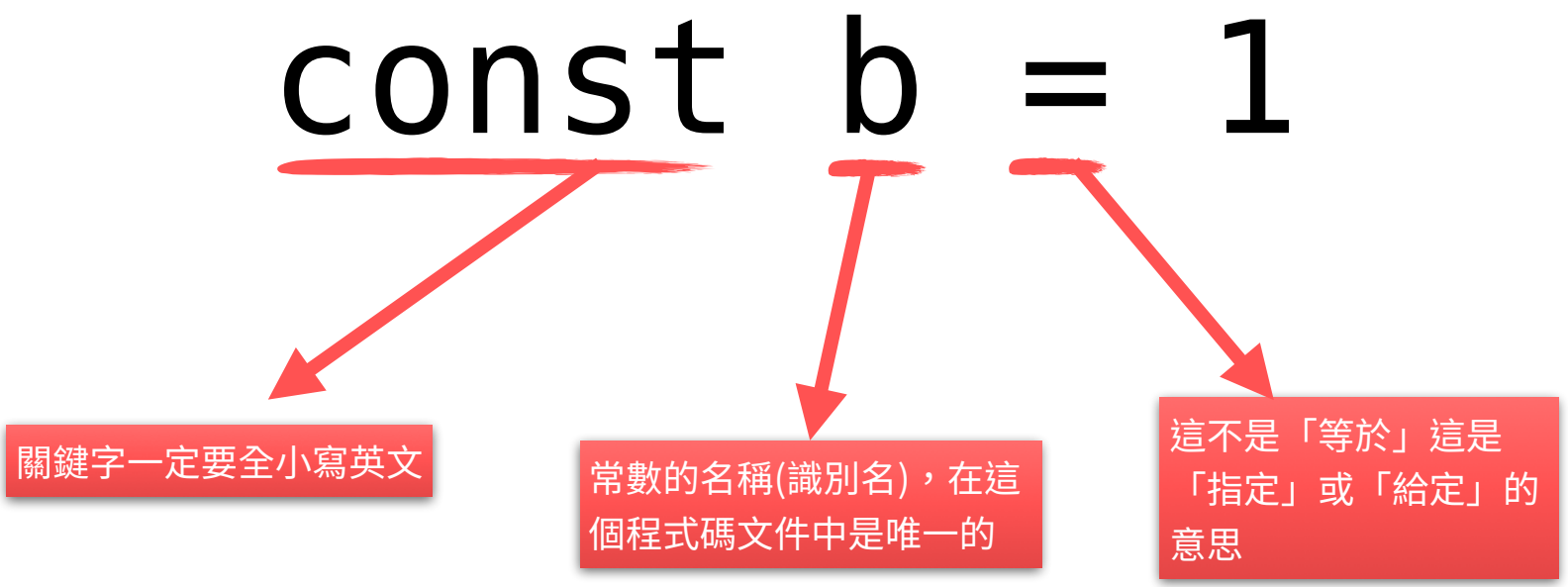
變數

變數的資料類型，會隨著給定的值而變動，這稱為「動態資料類型」或是「鬆散(弱)資料類型」的特性

```
let a = 1
a = 'hello'
```

常數

宣告時使用 **const** 關鍵字，常數規定宣告時一定要給定初始值，而且之後不能再次作給定運算



常數

常數是一種「**具有固定值的變數**」，一開始宣告時給定值後，就不能再次作給定的運算，所以它的資料類型一開始就決定好了

```
const b = 1
b = 'hello'
```

常數 b 的資料類型現在是「數字」

作再次指定的運算，這行會造成錯誤

常數

常數可以用於複合型的資料類型，例如 **物件** 或 **陣列** 的宣告，因為裡面記錄的是「**地址**」，類似於某大樓的地址，如果改變裡面的成員並不會更動到地址

常數

? 範例中的 d 陣列經過 c 陣列改變成員值後，此時裡面的成員是什麼？

```
const c = [1, 2]
const d = c
c[0] = 3
```

常數 c 是一個陣列，裡面有兩個成員

常數 d 指定為 c 陣列，d 的地址與 c 相同

改變 c 陣列中的成員值

常數

常數也可以用於函式表達式的宣告，函式表達式只能使用 **const** 宣告

```
const foo = function( ) { }
```

註：JS中的函式宣告有兩種，一種是函式定義語法(簡稱FD)，一種是函式表達式語法(簡稱FE)。
它們會被使用在不同的場合，而且某些特性不太相同。

常數

常數也可以用於函式表達式的宣告，函式表達式只能使用 **const** 宣告

```
const foo = function( ) { }
```

註：JS中的函式宣告有兩種，一種是函式定義語法(簡稱FD)，一種是函式表達式語法(簡稱FE)。
它們會被使用在不同的場合，而且某些特性不太相同。

變數&常數撰寫風格建議

- ✓ 優先使用常數宣告(const)
- ✓ 使用變數宣告(let)時，宣告時就要指定初始值，如果不確定是什麼類型的值，可以使用null
- ✓ 在函式或程式碼文件的最前面宣告變數(常數)
- ✓ 一行語句宣告一個變數(常數)
- ✓ 把let的宣告放在一起，const的宣告放在一起

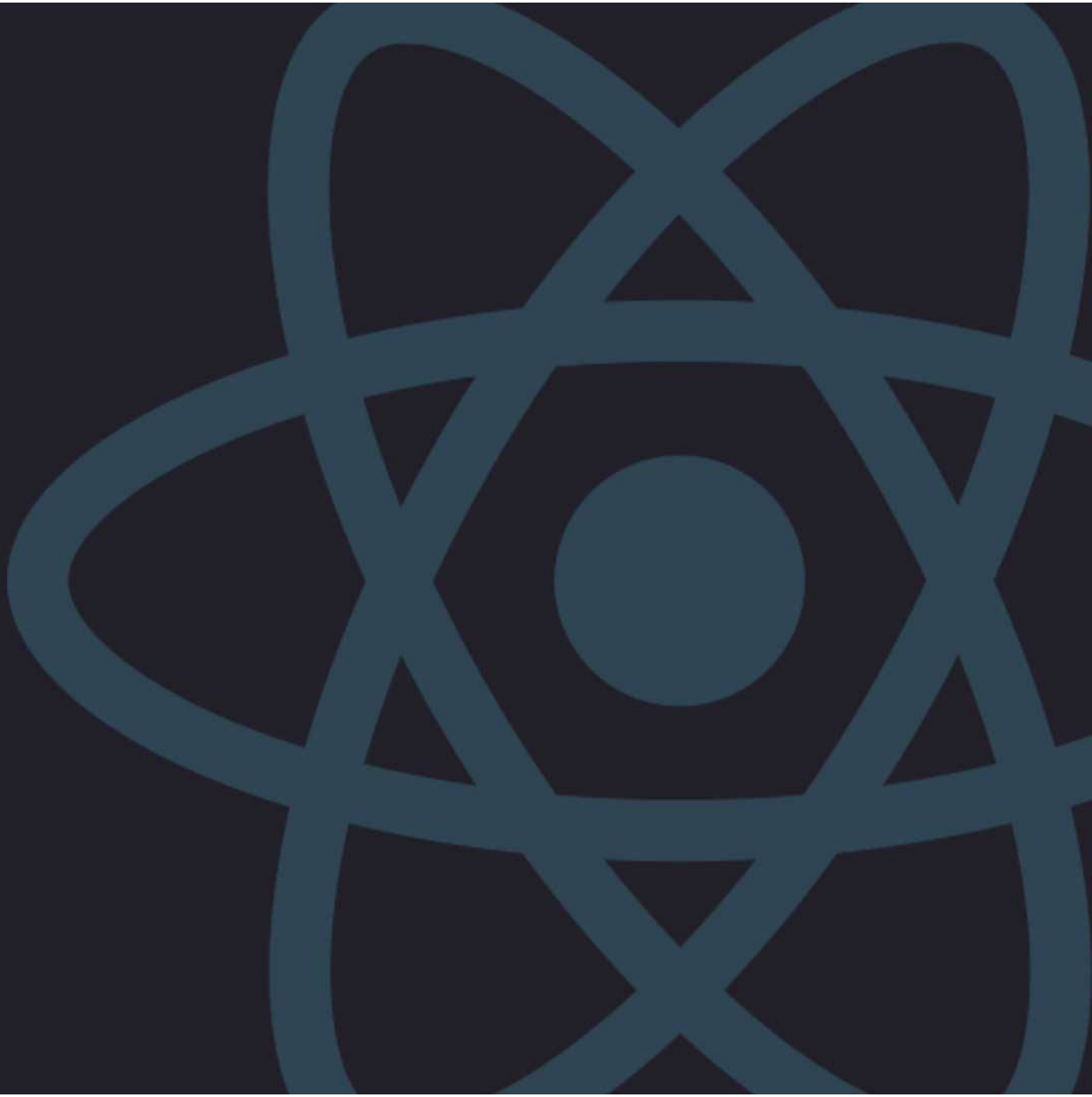
變數&常數進階議題

- ✓ 區域作用域(block) vs 函式作用域(function)
- ✓ for圓括號中的let變數仍然是在區塊作用域
- ✓ for迴圈中的let變數會作重新綁定
- ✓ let與const的提升(hoisting)，以及TDZ(暫時死區)

註：以上都是比較ES6之前，單只用var來宣告變數的情況。

JavaScript(ES6)入門

資料類型



資料類型 - 數字

數字 - 一般的數字值，浮點數(有小數點)與整數，有最大與最小極限值，浮點數運算也有精度問題

```
const a = 1
typeof a
1.0 === 1
0.1 + 0.2
```

資料類型 - 數字

數字 - 有一些特別的記號也是數字類型，正無限大、負無限大與不是數字。0在運算時會區分+0與-0

```
Infinity
-Infinity
NaN
```

資料類型 - 數字

數字 - Math與Number物件提供了數字相關屬性與方法，以及最大、最小值

```
Number.MAX_VALUE
Number.MIN_VALUE
Math.PI
Math.round(4.7)
```

資料類型 - 數字運算

數字 - 最基本的運算是「加減乘除餘」稱為數學運算。另外大於、小於、相等比較，數字也是較優先與常用的運算

```
1 + 1
2 > 1
3 === 3
3 == 3
```

資料類型 - 轉為數字

數字 - 從其它類型轉為數字，最簡單的方式是使用一元正號(+)

```
+'123'
+'123abc'
+true
+false
```

資料類型 - 字串

字串 - 使用單引號(')或雙引號(")框住的文字，兩者結果相同，但建議使用單引號(')，另有一種新式的樣版字串是使用重音符號backtick(``)

```
'123'
"a"
`true`
```

註: JavaScript中沒有「字元」這種資料類型

資料類型 - 字串運算

字串 - 使用加號(+)作串接運算，取出字串的一部份要使用slice方法，另外length屬性可得知字串的長度(字元個數)

```
'123'.length
'abcde'.slice(0,2)
'hello'+'world'
```

資料類型 - 字串運算

字串 - 字串串接運算 與 數字的相加運算，都是使用加號(+)。**字串串接運算較優先。**

```
'123' + 1
```

註: 要將其它資料類型的值轉為字串，最簡單的方式就是與一個空白字串相串接

資料類型 - 布林

布林 - 使用絕對兩分法的值(黑白/陰陽/真假)，true與false來作為布林的兩種可用的值。常用於流程控制的結構中。

```
true
false
```

註: JavaScript關鍵字都是全小寫的

資料類型 - 布林

假家族(falsy)　+0, -0, null, NaN, undefined, 空白字串(''), false

真家族(truthy) - 不是假家族，也就是有值的情況

資料類型 - 空值

null - 空值，給開發者用的。(typeof null === 'object')

undefined - 未定義，通常是系統或函式庫使用的。

註: JavaScript關鍵字都是全小寫的