

# #1

**let & const**

**變數&常數宣告**

**JS**

# 變數(Variable)

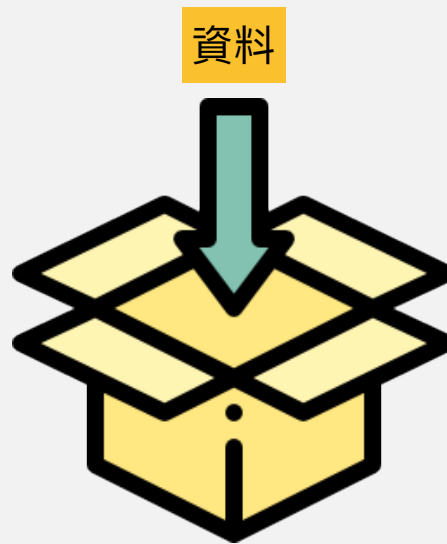
ABOUT US

INFOGRAPHICS

CONTACT

2

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



Icon made by Elias Freepik(<http://www.freepik.com/>) from [www.flaticon.com](http://www.flaticon.com)

# 變數(Variable)

ABOUT US

INFOGRAPHICS

CONTACT

3

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

let a = 1

關鍵字一定要全小寫英文

變數的名稱(識別名)，在這個程式碼文件中是唯一的

這不是「等於」這是「指定」或「給定」的意思

# 變數(Variable)

ABOUT US

INFOGRAPHICS

CONTACT

4

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'
```

變數 a 的資料類型現在是「數字」

變數 a 的資料類型現在是「字串」

# 常數(Constant Variable)

ABOUT US

INFOGRAPHICS

CONTACT

5

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

**const b = 1**

關鍵字一定要全小寫英文

常數的名稱(識別名)，在這個程式碼文件中是唯一的

這不是「等於」這是「指定」或「給定」的意思

# 常數(Constant Variable)

ABOUT US

INFOGRAPHICS

CONTACT

6

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(Constant Variable)

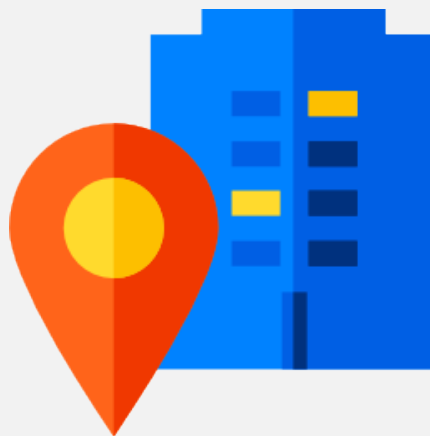
ABOUT US

INFOGRAPHICS

CONTACT

7

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



Icon made by Elias Freepik(<http://www.freepik.com/>) from [www.flaticon.com](http://www.flaticon.com)

# 常數(Constant Variable)

ABOUT US

INFOGRAPHICS

CONTACT

8

① 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(Constant Variable)

[ABOUT US](#)[INFOGRAPHICS](#)[CONTACT](#)

9

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 建議撰寫風格(習慣)

ABOUT US

INFOGRAPHICS

CONTACT

10

- ✓ 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

# #1

Data Type  
資料類型

JS

# JS中的基礎資料類型

ABOUT US

INFOGRAPHICS

CONTACT

12

- ✓ 基礎：數字(Number)、字串(String)、布林(Boolean)
- ✓ 空值或未定義：null、undefined
- ✓ 複合(集合)或參照：物件(Object)、陣列(Array)
- ✓ 特殊：符號(Symbol)

# 數字(Number)

[ABOUT US](#)[INFOGRAPHICS](#)[CONTACT](#)

13

數字(Number)並無分整數或浮點數，對JS來說，1.0與1是相同的。數字最重要的運算有加減乘除餘(+\*/%)等算術運算，以及大於(>)小於(<)等比較運算。

特殊的數字值：

- Infinity - 無窮大(不帶正負號時為正無窮大)
- NaN - 不是數字(Not a Number)。通常是運算或轉換後得到的值不是數字時
- Number.MAX\_VALUE
- Number.MIN\_VALUE

# 字串(String)

[ABOUT US](#)[INFOGRAPHICS](#)[CONTACT](#)

14

描述文字的資料類型，沒有區分字元或字串，用單引號(')或雙引號("")宣告完全相同。

```
let a = 'hello'  
let b = a + 'world'
```

字串的連結(concatenate)運算

# 加號(+)的符號共同情況

ABOUT US

INFOGRAPHICS

CONTACT

15

加號(+)常見的有三個符號共用的情況：

1. 數字相加算術運算
2. 字串連結運算
3. 正號(一元正號)

```
let a = 'hello'  
let b = a + 123
```

變數b的結果是什麼？

## 一元正號(+)

[ABOUT US](#)[INFOGRAPHICS](#)[CONTACT](#)

16

一元正號(+)通常是用來 **將其它類型的資料強制轉為數字類型**  
尤其是其中只包含數字的字串

```
let a = +'101'  
let b = +new Date( )
```