

| 為你自己學 系列課程

JavaScript 入門

課程代碼 JS101

{💎} 五倍紅寶石



資料型態

Data Type

```
let name = "Hello World"
```

變數本身沒有形態

資料型態

Data Type

87

3.1415926

"Hello World"

'五倍の選擇'

[1, 4, 5, 0]

{ name: "悟空", skill: "龜派氣功" }

true

false

undefined

null

function () { }



値

原始型別 vs 物件型別

Primitive Type vs Object Type

原始型別

- 數字 (Number)
- 字串 (String)
- 真假值、布林值 (Boolean)
- 空值 (Null)
- 未定義 (Undefined)
- 符號 (Symbol)

物件型別

- 陣列 (Array)
- 函數 (Function)
- 物件 (Object)

為什麼要有不同的型別？

怎麼看是什麼型別？

```
console.log(typeof 123) // 印出 "number"  
console.log(typeof 3.14) // 印出 "number"  
console.log(typeof "hello") // 印出 "string"  
console.log(typeof true) // 印出 "boolean"  
console.log(typeof false) // 印出 "boolean"  
console.log(typeof undefined) // 印出 "undefined"  
console.log(typeof Symbol("hello")) // 印出 "symbol"  
console.log(typeof null) // 印出 "object" !!!!
```

數字

Number

四則運算之優先順序

先乘除後加減

```
console.log(2 + 3 * 2); // 會印出什麼？  
console.log((2 + 3) * 2); // 會印出什麼？
```


`a++` vs. `++a`

```
var a = 1;
```

```
a++;
```

遞增

```
console.log(a); // 會印出什麼？
```

2

```
++a;
```

遞增

```
console.log(a); // 會印出什麼？
```

3

字串

String

真假值 / 布林值

Boolean

真的, 假的, 沒有其它

true or false

不同的型別之間是可以轉換的

型別轉換

```
let n = 3.14  
console.log(typeof n) // 印出 number
```

```
let m = String(n)  
console.log(typeof m) // 印出 string
```

```
let x = "1450"  
console.log(typeof x) // 印出 string
```

```
let y = Number(x)  
console.log(typeof y) // 印出 number
```


如果轉不動的話？

```
let name = "hello"
```

```
let friends = ["John", "Mary", "Helen"]
```

```
console.log(Number(name)) // 會印出什麼？
```

```
console.log(Number(friends)) // 會印出什麼？
```

NaN

Not a Number

```
console.log(typeof NaN) // 會印出什麼？
```

NaN 是個很特別的存在

如果把不同型別放在一起的話...

```
console.log(1 + 2) // 印出 3
```

```
console.log(1 + "2") // 印出 12
```

```
console.log("hello" + 123) // 印出 hello123
```

```
console.log("hello" + true) // 印出 hellotrue
```

```
console.log(123 + true) // 印出 124
```

```
console.log(123 + false) // 印出 123
```

強制轉型

Type Coercion

【 冷^カ知^チ識^シ 】

// 猜猜看以下這 4 行會印出什麼？

```
console.log([] + [])
```

```
console.log([] + {})
```

```
console.log({} + [])
```

```
console.log({} + {})
```