

JavaScript @ AstroCamp

Part 1: JavaScript 基本語法

用瀏覽器打開 **https://jsbin.com**

- >> 打開 JavaScript 及 Console 標籤
- >> 下拉 JavaScript 選單，改成 ES6/Babel
- >> 取消 HTML 及 Output 標籤

最基本的東西（資料）：

>> 數字

>> 字串

>> 布林值

基本值 (primitive)

// Number 數字

1 // Integer 整數

0.3 // Float 浮點數

// String 字串

"foo"

'bar'

`baz`

// Boolean 布林值

true

false

數字：加減乘除及其它

// 運算子 (Operator)

1 + 1;

10 - 1;

3 * 5;

9 / 4;

10 % 2;

3 ** 4;

// Math 函式 (Math function)

Math.round(5.5);

Math.min(10, 3, 9);

布林值：對或是不對

true

false

很少直接寫，但有些函式的回傳值會是這種東西

為什麼要有變數

變數宣告 (Declaring variables)

```
let a = 1; // 區塊變數
```

```
console.log(a + 2);
```

```
const b = 2; // 常數
```

```
b = 3; // 會噴錯誤
```

```
// 常見，舊版
```

```
var c = 4;
```

```
c = c + 1;
```

```
console.log(c);
```


所謂的慣例是什麼？

- >> 變數名稱取法
- >> 程式檔名取法
- >> 函式的順序
- >> 檔案與資料夾結構

到底要不要加分號？

一次宣告兩個變數

```
let x = 10, y = 20;
```

```
// 等同於下面兩行
```

```
// let x = 10;
```

```
// let y = 20;
```

```
console.log(x + y);
```

裝邏輯的小盒子：函式

```
function greeting(name) {  
  // say a friendly hi  
  console.log("Hello world! This is" + name)  
  return `name: ${name}`  
}
```

```
let ret = greeting('tai')  
console.log(ret);
```

每天都會用到的東西

`/*`

區塊註解 Block comments

`*/`

`//` 單行註解

`console.log("Hello world!")` `//` 把東西印出來看

定義一個叫 `foo` 的函式 (function)

```
function foo(){  
  // do something  
  return 1 + 1  
}
```

函式名稱只能用英文字母、底線 `_` 或錢字號 `$` 開頭，接著可以有數字(但很少用)。拼寫的慣例上會用駝峰字 (camelCase)。例如 `giveMeYourName`。

註：在其它語言 (Ruby, Python) 裡有個詞叫“方法”，一樣的概念，但有些許不同。

"呼叫"這個函式

foo();

呼叫、調用都是常用的動詞，英文則會用 `call`、`invoke`。基本上就是執行函式。

接收一個參數的函式

```
function printAddOne(x) {  
    console.log(x + 1);  
}
```

```
printAddOne(10);
```


回傳值

```
function addOne(x) {  
    return x + 1;  
}
```

```
addOne(10);
```

注意：在 `return` 之後的程式碼都不會被執行到。

練習一下

接收兩個參數的函式

```
function add(x, y) {  
    return x + y;  
}
```

//TODO: 要怎麼呼叫這個函式呢？

重要小工具

DevDocs 或 Dash (+ Alfred)

回到基本值

// 縮寫

c++ // c = c + 1

c--

// 更多縮寫

c += 2;

c -= 3;

c *= 4;

c /= 2;

c %= 2;

比較兩個東西

3 > 2

2 >= 0

5 < 5

6 <= 10

這兩個是一樣的東西嗎？ == 與 ===

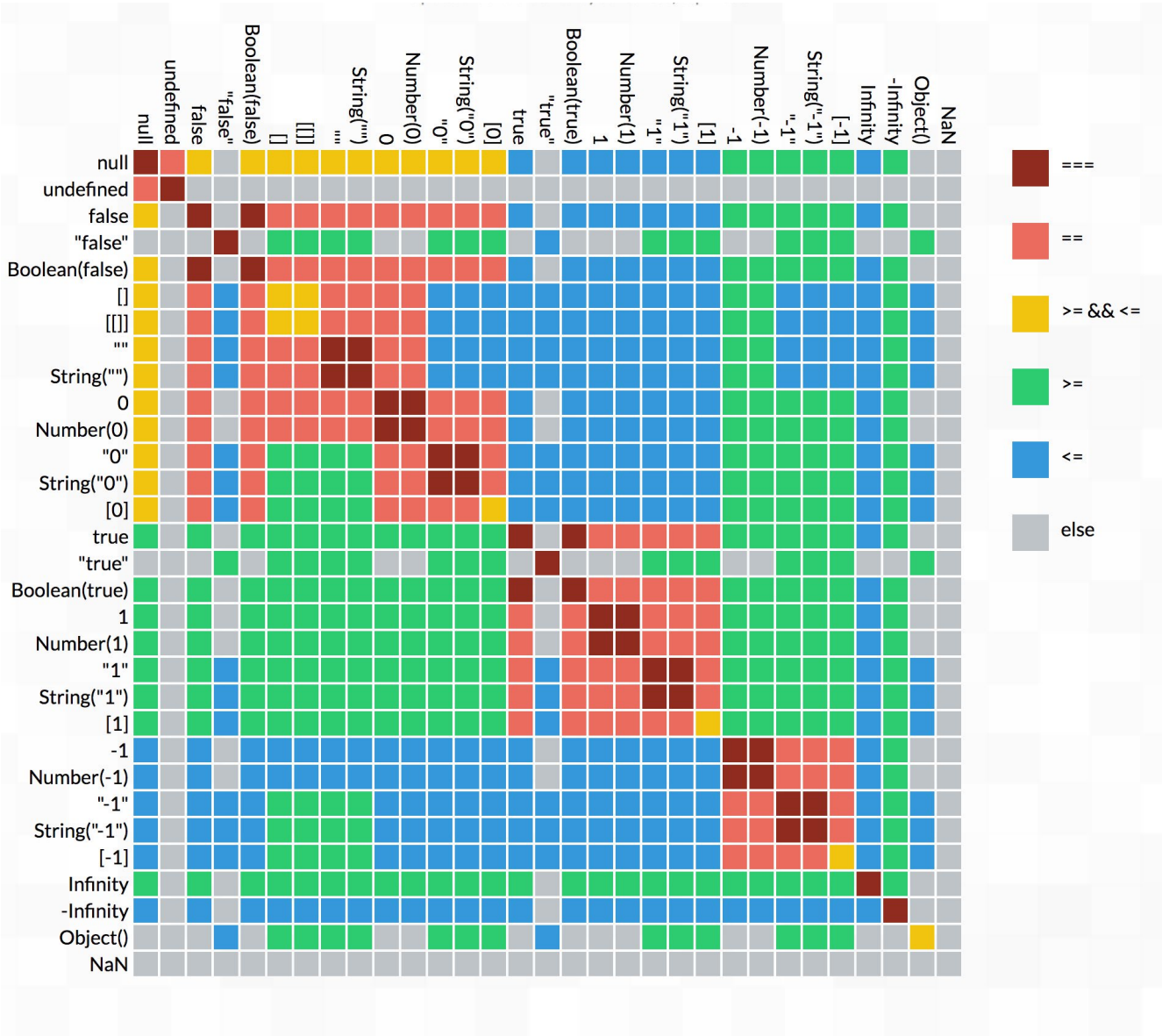
```
let numberOne = 1
```

```
let stringOne = "1"
```

```
console.log(numberOne == stringOne)
```

```
console.log(numberOne === stringOne)
```

惡名昭彰的表



還存在的東西

```
console.log(ghost);
```

```
let anotherGhost = null;  
console.log(anotherGhost);
```

字符串：串接

```
let userName = 'Chris';  
let greeting = 'Hello, ' + userName + '! Nice to meet you.'  
console.log(greeting)
```

字串：interpolation

```
let greeting = `Hello ${userName}! Nice to meet you.`
```

條件分支 (conditionals) : if / else

```
if (110 > 100) {  
    console.log("lalala")  
}
```

```
let cartAmount = 30
if (cartAmount >= 100) {
    cartAmount = cartAmount * 0.85
} else {
    cartAmount = cartAmount * 0.9
}
```

超過兩個條件怎麼辦？

```
let cartAmount = 30;
if (cartAmount >= 1000) {
  cartAmount = cartAmount * 0.8;
} else if (cartAmount >= 100) {
  cartAmount = cartAmount * 0.85;
} else {
  cartAmount = cartAmount * 0.9;
}
```

練習一下

有沒有辦法寫的更漂亮一點？

Tips: 有哪些部份是重複的？

條件分支：case

```
let gender = 'F', title;  
switch(gender) {  
  case 'M':  
    title = 'Mr.';  
    break;  
  case 'F':  
    title = 'Ms.';  
    break;  
  default:  
    title = '';  
}
```

```
console.log(title);
```


練習一下

集合 Collections

[] 跟 {}

1. 丸子串：陣列

```
let genre = ['pop', 'jazz', 'tango', 'blues']
```

>> 通常是一群類似的東西依順序放在一起

2. 像是字典的東西：鍵值對

```
let fruitPrices = {"apple": 3, "banana": 2, "orange": 4}
```

key-value pair 在其它的語言裡有不同的名字，Ruby 裡叫 Hash、Elixir 裡叫 Map、Java 裡叫 HashMap、Python 跟 C# 裡叫 Dict(dictionary)。在 JavaScript 裡叫 Object(先忘記其它的語言上的 Object)。

物件的兩種用法

1. 查表
2. 表示一個實體的屬性

匿名函式

居然可以在這個地方宣告函式？

作業：

1. 把 nodejs 環境設好

// 輸入會是一個數字，回傳一個陣列，從 0 開始到你輸入的數字結束，除了以下幾種：

// 3 的倍數要換成字串 Fizz

// 5 的倍數要換成字串 Buzz

// 又是 3 又是 5 的倍數要換成 FizzBuzz

// 例如輸入 6

// 會回傳 [1, 2, Fizz, 4, Buzz, 6]

``js

function fizzBuzz(input) {

//把你的解答寫在這裡。要記得 return 喔。

}

Happy hacking!