Rails Girls Weekly

JavaScript Basic

Week 4

2015-10-08 @ 五倍紅寶石出礦坑

Speaker: Win Wu

Agenda

- **Array** 陣列
- **Object** 物件
- **JSON**

Array 陣列

陣列在 **JavaScript** 是一種資料型態

Array 陣列

陣列讓我們把很多資料一次存在同一個變數裡

建立 **Array** 的方式可以

用 **Array Literal** 或是 **new Array ()** 的方式

較不建議使用 **new Array()** 的方式，參考: **http://www.w3schools.com/js/js_arrays.asp**

Array 陣列

Array Literal

每個資料都要以 , 隔開，並且用 [ ] 中括號把資料包起來

```
var arrayName = [item1, item2, ...];
```

Array 陣列

new Array()

用 **new** 關鍵字去建立一個 **Array**

var arrayName = new Array(item1, item2);

Array 陣列

兩種建立 **Array** 的方式範例:

```
var demo = ['王小明', '黃小美', 'winwu'];
console.log(demo);

var demo2 = new Array('王小明', '黃小美', 'winwu');
console.log(demo2);
```

http://jsbin.com/ginepuyubu/edit?js,console,output

Array 陣列

取得陣列的某個值

注意 **JavaScript** 的 **index** 是從 **0** 開始算哦！

arrayName[第幾個];

```
var demo = ['王小明', '黃小美', 'winwu'];

// 取得 demo 陣列中第三位同學
var whoIsThird = demo[2];
console.log(whoIsThird);
```

Array 陣列

更改陣列的某個值

arrayName[第幾個] = 新的值;

```
var demo = ['王小明', '黃小美', 'winwu'];

// 換掉第三位同學為 YIYING WU
demo[2] = 'YIYING WU';
var whoIsThird = demo[2];
console.log(whoIsThird);
```

Array 陣列

得知陣列有幾個項目

arrayName.length

Array 陣列

```
var contacts = ['王小明', '黃小美', 'winwu', 'amy', 'allen',
'Iris', 'Irene', 'steven', 'Rock', 'Leo', 'LEON', 'Joy','王小明',
'黃小美', 'winwu', 'amy', 'allen', 'Iris', 'Irene', 'steven',
'Rock', 'Leo', 'LEON', 'Joy','王小明', '黃小美', 'winwu', 'amy',
'allen', 'Iris', 'Irene', 'steven', 'Rock', 'Leo', 'LEON',
'Joy'];

/*
 * 用 arrayName.length 可以知道 arrayName 這個陣列
 * 總共有幾個項目
 */

var totalPeople = contacts.length;
console.log(totalPeople);
```

Array 陣列總結

- 記住 **Array** 的 **index** 是從 **0** 開始
- 要注意陣列是用 **[ ]** 包起來
- 陣列可以使用的方法**(method)**很多，像是 **map, join, push, pop, reduce, reverse, slice, shift, toString …**

Object 物件

物件在 **JavaScript** 是一種資料型態

我們用大括弧來表示

Object 物件

宣告一個變數型態為 **Object** 會像這樣:

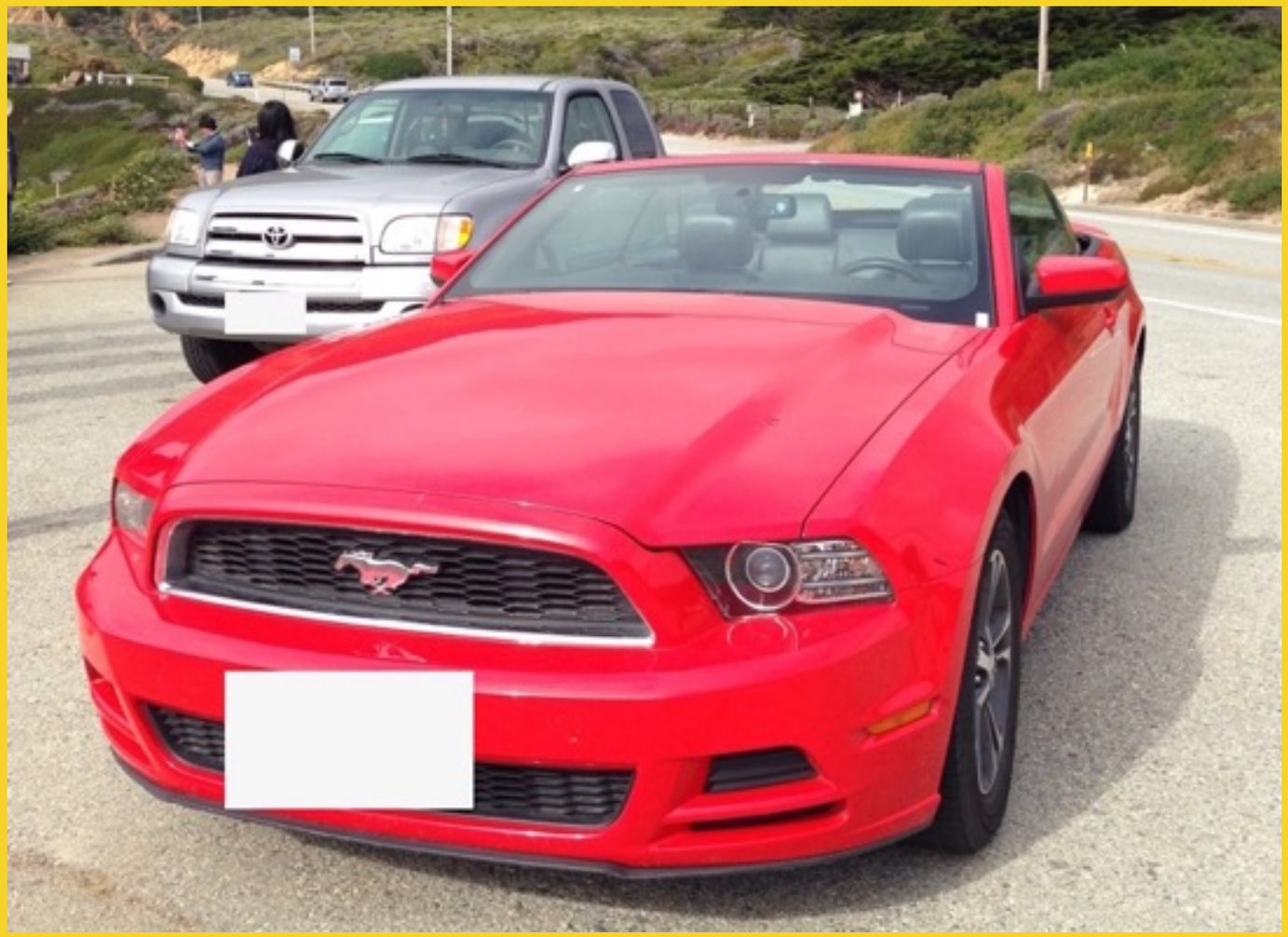
```
var objectName  = { };
```

Object 物件

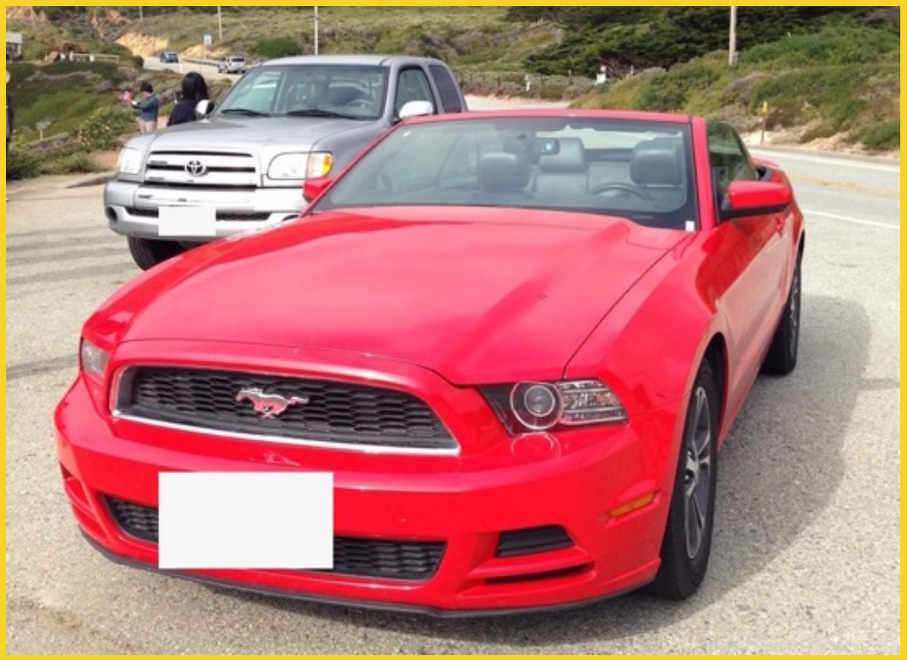
物件是什麼?!

Object 物件

真實世界中的物件，比方說車子，就是一個物件。



Object 物件



```
var car = { };
```

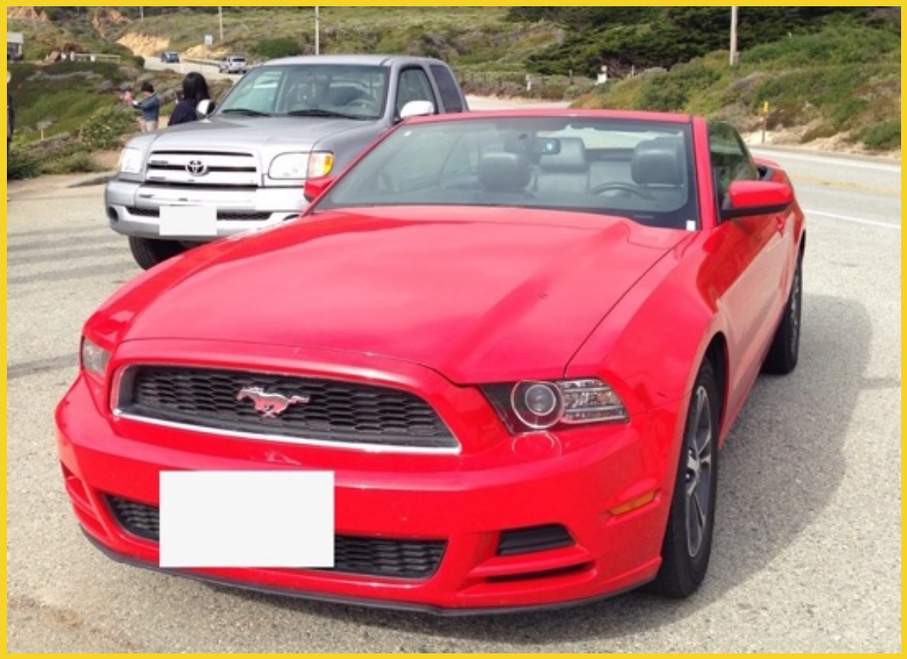
Object 物件

```
var car = { };
```

我們需要為這台車子描述更多關於他本身的事情
讓這個物件的變數更加有意義。

Object 物件

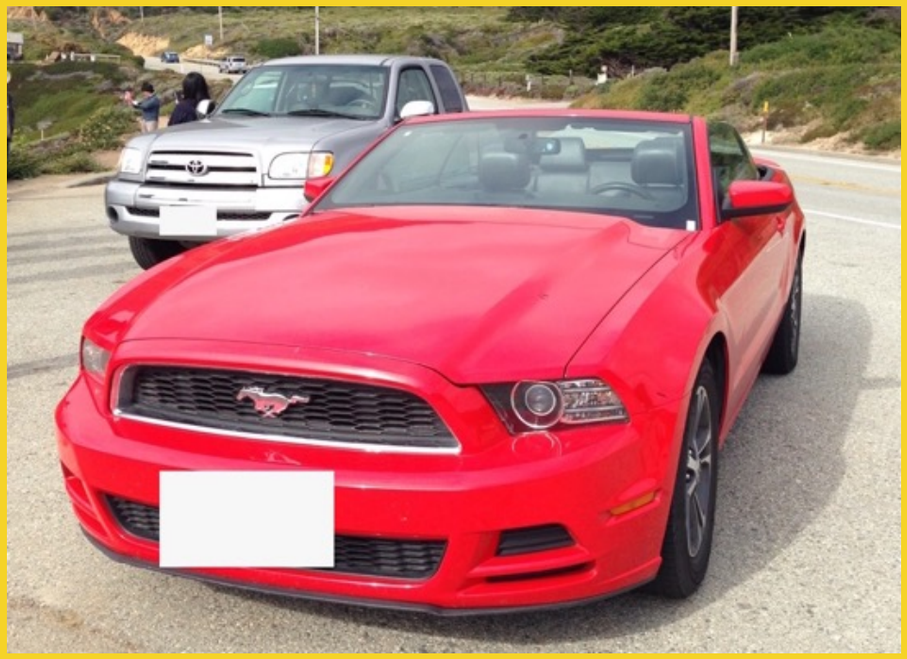
觀察這台車子的『屬性』



- 顏色
- 品牌
- 最大馬力
- 排氣量
- 長
- 寬
- 高
- 車重
- …

Object 物件

觀察這台車子的『方法』



- 發動
- 煞車
- 熄火
- 刷雨刷
- …

Object 物件

表達這個物件的『屬性』跟『方法』
將屬性跟方法的定義放在 **{ }** 裡面
並且要用 **key : value** 的方式定義
每組『**key:value**』要用『**,**』隔開

```
var objectName = {
   propertyName: value,
   methodName: method
};
```

Object 物件

用程式表達這個車子物件的『屬性』跟『方法』

Practice:

```
var car = {
  color: "red",
  type: "Microcar",
  age: '2'
};
```

屬性名稱 (Property)	屬性值
color	red
type	Microcar
age	2

Object 物件

```
var car = {
  color: "red",
  type: "Microcar",
  age: '2',
  start: function() {
    console.log('車子發動!');
  },
  stop: function() {
    console.log('車子熄火');
  },
};
```

取得物件的屬性值

有兩種方法可以存取物件的屬性值:

(1) objectName.propertyName

(2) objectName[“propertyName"]

例如: 取得 **car** 的 **color** 屬性值:

```
> car.color
<· "red"
> car["color"]
<· "red"
```

取得/呼叫物件的方法

呼叫物件的方法:

objectName.methodName()

取得物件的方法 (定義):

objectName.methodName

```
> car.start()
  車子發動!
< undefined
> car.start
< function () {
      console.log('車子發動!');
  }
```

定義物件的一種方式

用大括弧{ }，
並且使用一串用逗號,分隔的 **key:value** 對組
所定義物件的方式叫做物件字面值 **(Object Literal)**

物件總結

- **key** 不能重複使用
- **value** 可以是各種型別的值 **(Boolean, string, integer, object, function…)**
- 定義物件的三種方式:
  - 物件字面值 **(Object Literal)**
  - 使用 **new** 關鍵字建立物件
  - **ECMAScript5** 的 **Object.create()**

http://jsbin.com/pewefifula/edit?js,console

Array 跟 Object 有什麼不同?

Array is **numbered** index.

Object is **named** index.

JSON

JavaScript Object Notation

是由 **Douglas Crockford** 構想設計的

輕量級資料交換語言格式。

JSON

JSON 是獨立於語言的文字格式。

JSON 也使用於其他程式語言，大部份與網頁開發相關的語言都有 **JSON** 函式庫 (像是 **php** 的 **json_encode, json_decode)** 。

JSON 常見的使用情境

一般 **JSON** 在 **web** 開發上常用來做前後端的資料交換，前端發送 **Request** 給後端請求資料 **(**通常透過 **AJAX)**，不論是取得或是發送的資料格式都可以使用 **JSON** 資料格式 **(**先約定**)**，接著就可以解析 **JSON** 格式裡的資料做資料呈現等等。

如果你會使用 **jQuery** 的話，**getJSON** 跟 **parseJSON** 都是常用的解讀函式。

JSON 常見的應用

- 存取 **API**
- **Single Page Application (AJAX 頁面)**

http://www.w3schools.com/js/js_json.asp

如何建立 JSON 字串

- **JSON** 字串可以包含陣列 **Array** 或是物件 **Object**
  - 陣列可以用 **[ ]** 來寫入資料
  - 物件可以用 **{ }** 來寫入資料，每組 **"key" : "value"** 用 **,** 區隔。名稱／值（**collection**）：名稱和值之間使用「:」隔開，大致上跟我們平常寫 **Object** 一樣的格式，只是內容都是字串。

http://www.w3schools.com/js/js_json.asp

JSON 格式範例

http://www.json-generator.com/

JSON 格式檢查

http://jsonlint.com/

- **JSON** 格式沒有註解，他不是程式語言，他只是一種資料交換格式
- 如果想要將 **JSON** 格式的資料獨立在一隻檔案裡，副檔名為 **.json**

javascript 如何解讀 JSON

week4/index.html

- JSON 格式轉 object物件來操作 :
  - JSON.parse (JSON格式資料)
    - 會回傳給你陣列或是物件

- Object 轉成 JSON 字串格式 :
  - JSON.stringify (object)
    - 將 JavaScript 值轉換成以 JavaScript 物件標記法 (JSON) 表示的字串。

http://www.w3schools.com/js/js_json.asp

遇到瀏覽器不支援 **JSON?** 找 **json2.js**

https://github.com/douglascrockford/JSON-js

Reference

- http://www.w3schools.com/js/js_objects.asp
- http://blog.wu-boy.com/2011/04/%E4%BD%A0%E4%B8%8D%E5%8F%AF%E4%B8%8D%E7%9F%A5%E7%9A%84-json-%E5%9F%BA%E6%9C%AC%E4%BB%8B%E7%B4%B9/
- http://www.json.org/json-zh.html
- http://www.json.org/
- https://msdn.microsoft.com/zh-tw/library/cc836466(v=vs.94).aspx

Thank you :D