

JavaScript

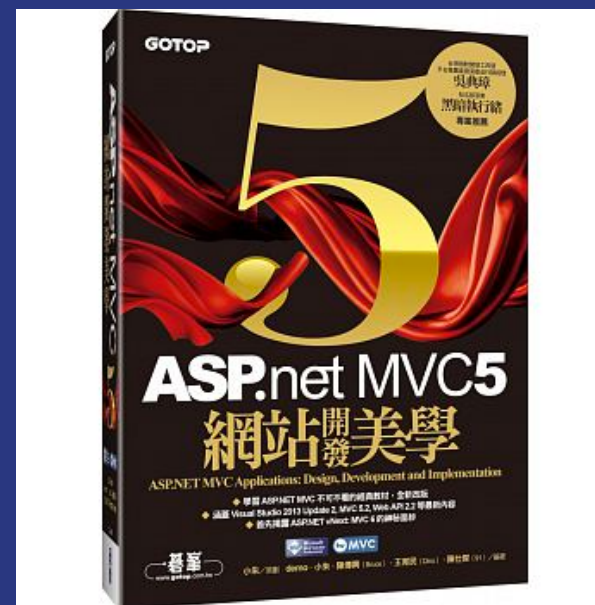
Dino Wang
Build School 講師

請尊重講師的著作權及智慧財產權!

Build School 課程之教材、程式碼等、僅供課程中學習用、請不要任意自行散佈、重製、分享，謝謝

講師介紹 – Dino Wang

- 微軟最有價值專家 – Azure
- twMVC Community Co-funder
- ASP.NET MVC 5 開發美學共同作者
- hexdigits
 - dino@hexdigits.com



生命靈數

- 使用 Bootstrap 完成本程式的簡單佈局
- 運用 JavaScript 的計算能力得到結果
- 結合 DOM 的操作，將結果輸出到畫面上

生命靈數實作中運用到的 JavaScript 特性介紹

- Date 物件
- JSON (JavaScript Object Notation)

Date 物件

- Date 物件用於表示特定的日期以及時間
 - 概念類似 C# 中的 DateTime 物件，但是使用方法各異
- 使用的方式，是以 new 保留字產生 Date 物件的實體

`new Date()` // 取得現在日期時間

`new Date("2017/3/1")` // 取得用以描述 2017/3/1 的 Date

以 Date 物件進行日期時間的取值

```
var now = new Date("2000/1/1");    // 解析日期文字  
var year = now.getFullYear();      // 回傳西元年  
var month = now.getMonth();        // 回傳月份(0~11)  
var date = now.getDate();           // 回傳日期
```

JSON (JavaScript Object Notation)

- JavaScript 物件表示法，是一種非常輕巧易懂的資料表示方式，可以簡單的語意來描述各種(資料)物件
- 由於其十分簡單的格式，也極為適合網路傳輸的場景，應用於網頁、手機、甚至 IoT 裝置
- <https://zh.wikipedia.org/wiki/JSON>

JSON (JavaScript Object Notation)

```
{  
  "name": "Hello Kitty",  
  "gender": "female",  
  "age": 41,  
  "marriage": false,  
  "weight": 5, "height": 3, "unit": "apple",  
  "friends": ["Daniel Dear", "Joy", "Fifi"]  
};
```

JSON (JavaScript Object Notation)

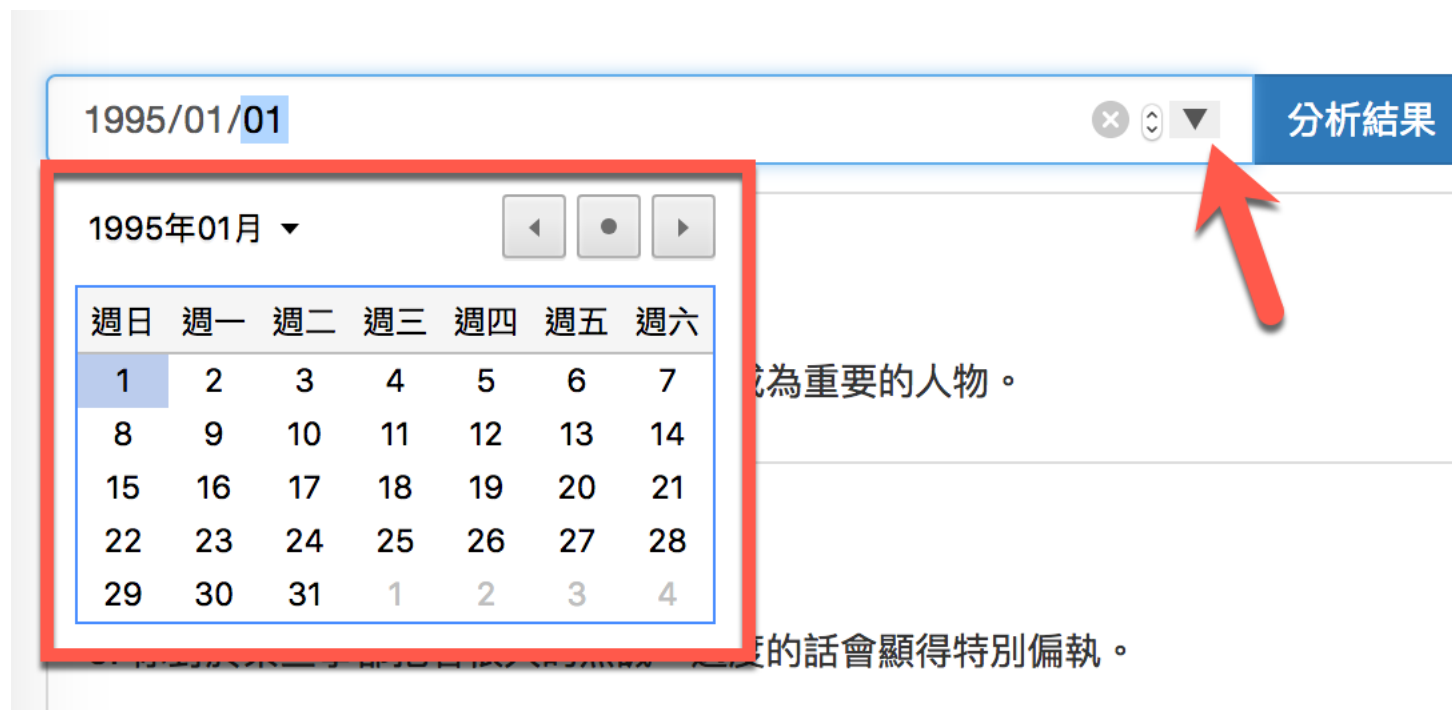
```
{  
  key "name": "Hello Kitty", value  
  "gender": "female",  
  "age": 41,  
  "marriage": false,  
  "weight": 5, "height": 3, "unit": "apple",  
  "friends": ["Daniel Dear", "Joy", "Fifi"]  
};
```

JSON (JavaScript Object Notation)

```
var idol = {  
    "name": "Hello Kitty",  
    "gender": "female",  
    ... 中略 ...  
    "friends": ["Daniel Dear", "Joy", "Fifi"]  
};  
var x = idol.name;           // 得到 Hello Kitty  
var y = idol["gender"];      // 得到 female  
var z = idol["friends"][0];  // 抓到第一位朋友
```

HTML 的補充 – 使用瀏覽器內建日期下拉

```
<input id="birthdate" class="form-control" type="date" name="birthdate" value="0">
```



1995/01/01

1995年01月

週日	週一	週二	週三	週四	週五	週六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

分析結果

成為重要的人物。

的話會顯得特別偏執。

提示解題步驟

- 根據輸入的日期，算出星座名稱
- 根據名稱，查表取得對應的星座資料
 - 取得星座中文顯示名稱
 - 算出生命靈數，並作為 索引值 查表取得講評
- 顯示於畫面上

Demo

檔名 horoscope-complete.html

舉例輸入 1995/02/01

- 透過函式轉換日期取得 "aquarius"
- 到 horoscopeData 中查到對應資料
 - horoscopeData["aquarius"]
- 由取得的資料中，拿出中文名稱
- 計算生命靈數
- 由取得的資料中查表，得到評語
- 顯示結果

```
var horoscopeData = {  
  "aquarius": {  
    "name": "水瓶座",  
    "lookup": [  
      "你是瓶子座中最自閉的，有時候會顯得太過我行我素了些。",  
      "你會比較受到別人的影響，但還不會失去自己原本的主張。",  
      "你是一隻多才多藝的瓶子哩！尤其擅長於表達與溝通哦！",  
      "你是瓶子中比較穩定的，會習慣維持某一種生活的方式。",  
      "常常讓人找不到的就是你這種瓶子啦！朋友也是多不勝數。",  
      "你是比較重感情的瓶子，也比較桃花，感情問題會比較多。",  
      "你是瓶子中最為理性的，太過聰明的話有時候會不近人情。",  
      "你是瓶子中比較重視物質的，會去追求自己所想要的生活。",  
      "你的理想非常高遠，如果持續努力，成功的機會是很大的。"  
    ]  
  },  
  "pisces": {  
    "name": "雙魚座",  
    "lookup": [  
      "你容易沉溺在自己的世界的，忘掉了身邊還有別人存在。",  
      "你很容易受到別人的引導，而忘記了自己原本的方向哦！"    ]  
  }  
}
```

1995-02-01 出生的你，星座是 水瓶座

你的生命靈數：

9. 你的理想非常高遠，如果持續努力，成功的機會是很大的。

1995/02/01

分析結果

Exercise

檔名 horoscope.html (有準備好的星座資料)

改善使用者體驗

- 這是一開始

Project name



1995-01-01 出生的你，星座是 摩羯座

你的生命靈數：

8. 對你來說功成名就吸引力大，會努力成為重要的人物。

1995/01/01



分析結果

改善使用者體驗

- 多查個幾筆就壞了 ...
- 輸入區整個跑掉

Project name 

1995-01-01	出生的你，星座是 摩羯座
你的生命靈數： 8. 對你來說功成名就吸引力大，會努力成為重要的人物。	
1995-02-01	出生的你，星座是 水瓶座
你的生命靈數： 9. 你的理想非常高遠，如果持續努力，成功的機會是很大的。	
1995-03-01	出生的你，星座是 雙魚座
你的生命靈數： 1. 你容易沉溺在自己的世界的，忘掉了身邊還有別人存在。	
1995-04-01	出生的你，星座是 牡羊座
你的生命靈數： 2. 你會在在意別人看你的眼光，個性比較害羞一點。	
1995-05-01	出生的你，星座是 金牛座
你的生命靈數： 3. 你是牛座中比較靈活的，喜歡發表自己所研究出來的事物。	
1995-06-01	出生的你，星座是 雙子座
你的生命靈數： 4. 你是比較重視生活規律的人，看來隨和但自有一套規則。	
1995-07-01	出生的你，星座是 巨蟹座
你的生命靈數： 5. 你不一定喜歡亂跑，但不能受到太多拘束，想自由自在。	



改善使用者體驗

- 多查個幾筆就壞了 ...
 - 輸入區整個跑掉
- 改善方式
 - 固定輸入框位置於頂端
 - 每次查詢結果顯示於列表第一筆

1995/01/01



分析結果

1995-01-01 出生的你，星座是 摩羯座

你的生命靈數：

8. 對你來說功成名就吸引力大，會努力成為重要的人物。

1995-02-28 出生的你，星座是 雙魚座

你的生命靈數：

9. 你對於某些事都抱著很大的熱誠，過度的話會顯得特別偏執。

1995-03-19 出生的你，星座是 雙魚座

你的生命靈數：

1. 你容易沉溺在自己的世界的，忘掉了身邊還有別人存在。

DOM 操作的變化

- appendChild 是從父節點後方加入新的元素，所以我的使用者介面受到一些限制，那有沒有辦法從前面加入呢？

DOM 操作的變化

- 有的，可以使用 insertBefore 方法，將新加入的元素放到指定的元素前
- `insertBefore(newChild, refChild);`
 - newChild 新加入的元素
 - refChild 參考的元素

https://www.w3schools.com/jsref/met_node_insertbefore.asp

insertBefore 範例

```
var list = document.getElementById("results");
```

```
var newChild = document.createElement("li");
```

```
newChild.className = "...";
```

```
newChild.innerHTML = "...";
```

```
if (list.childNodes.length > 0) {  
    list.insertBefore(newChild, list.childNodes[0]);  
} else {  
    list.appendChild(newChild);  
}
```

如果有任何的子節點存在

第一個子節點作為參考位置

改善使用者體驗

- 順便將輸入日期後按下 Enter，直接開始計算的功能也加入到生命靈數中

Demo and exercise

檔名 horoscope.html

小瑪莉轉盤



小瑪莉轉盤

- 使用 Bootstrap 完成本程式的畫面佈局
- 運用 JavaScript 的計算能力得到結果
- 結合 DOM 的操作，反覆操作畫面上已存在的元件

小瑪莉實作中運用到的 JavaScript 特性介紹

- 特殊的 if 判斷式行為
- 計時器
- 陣列 (一維陣列)
- 動態修改已存在 DOM 元素的屬性

特殊的 if 判斷式行為

- if (條件為真) { ... }
- 條件為真
 - true
 - 1
 - "1"
 - "Build-School"
 - ...

特殊的 if 判斷式行為

- 其實應該反過來舉例才比較完整，哪種結果不為真？
 - false
 - 0
 - ""
 - null
 - undefined
 - NaN

特殊的 if 判斷式

```
var brick = document.getElementById("i32");  
  
if (brick)    // 如果沒有取得物件 brick 不會有值  
{  
    // do something  
}
```

計時器

- JavaScript 提供四個函式作為計時器工作用途
 - `setTimeout`、`clearTimeout`
 - `setInterval`、`clearInterval`
- `setTimeout`
 - 於若干時間後執行一次指定函式
- `setInterval`
 - 反覆於若干時間後執行指定函式

計時器的使用方式

```
var tmr = setTimeout(sayHello, 1000);
```

```
function sayHello () {  
    alert("Hello~");  
}
```

```
//-----
```

```
var tmr = setTimeout(function () { alert("Hello~"); }, 1000);
```


計時器的使用方式

```
var tmr = setInterval(updateClock, 1000);
```

```
function updateClock () {  
    // 顯示時鐘  
}
```

```
document.getElementById("stop-button").onclick = function () {  
    clearInterval(tmr);  
};
```

Demo and exercise

檔名 777-slot.html

改善用戶體驗 (請自理)

- 游標加減速效果
 - 用得到數學了
- 積分計算
- 其他你對小瑪莉的想像

累進稅率

- 使用 Bootstrap 完成本程式的簡單佈局
- 運用 JavaScript 的計算能力得到結果
- 結合 DOM 的操作，將結果輸出到畫面上

累進稅率實作中預計到的 JavaScript 特性介紹

- JSON
- 無條件進位法

累進稅率實作中預計到的 JavaScript 特性介紹

- JSON
- 無條件進位法
 - Math 物件

提示：查表應用

- 該級稅率最低門檻

```
var taxRateLookup = [  
  { income: 10310001, rate: .5 },  
  { income: 4530001, rate: .4 },  
  { income: 2420001, rate: .3 },  
  { income: 1210001, rate: .2 },  
  { income: 540001, rate: .12 },  
  { income: 0, rate: .05 },  
];
```

Project name



累進稅率

50000000

計算稅額

總收入: 2000000, 稅率: 0.2, 應繳: 400000

總收入: 50000000, 稅率: 0.5, 應繳: 25000000

Exercise

檔名 tax-rate-level.html

在這裏你將學到

Learn How to Learn

- 學新東西、新技術的能力
- 尋找解答的能力
- 隨時吸取新知識的能力

跟著你一輩子的能力 ...

對學生的好處

不只是就業銜接，培養自我解決問題的能力及信心！



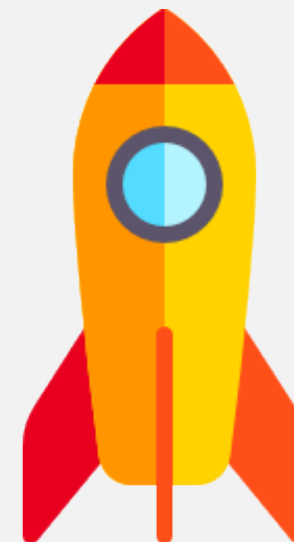
解決問題能力 (Problem-Solving)

- Learn how to learn
- Peer Learning
- Project-based



就業銜接 (Professional skills ready)

- Technical + Soft Skill



擴散及影響 (Make impacts)

- Mentor Networks

Q&A

課程資料 – <http://xxx.xxx>