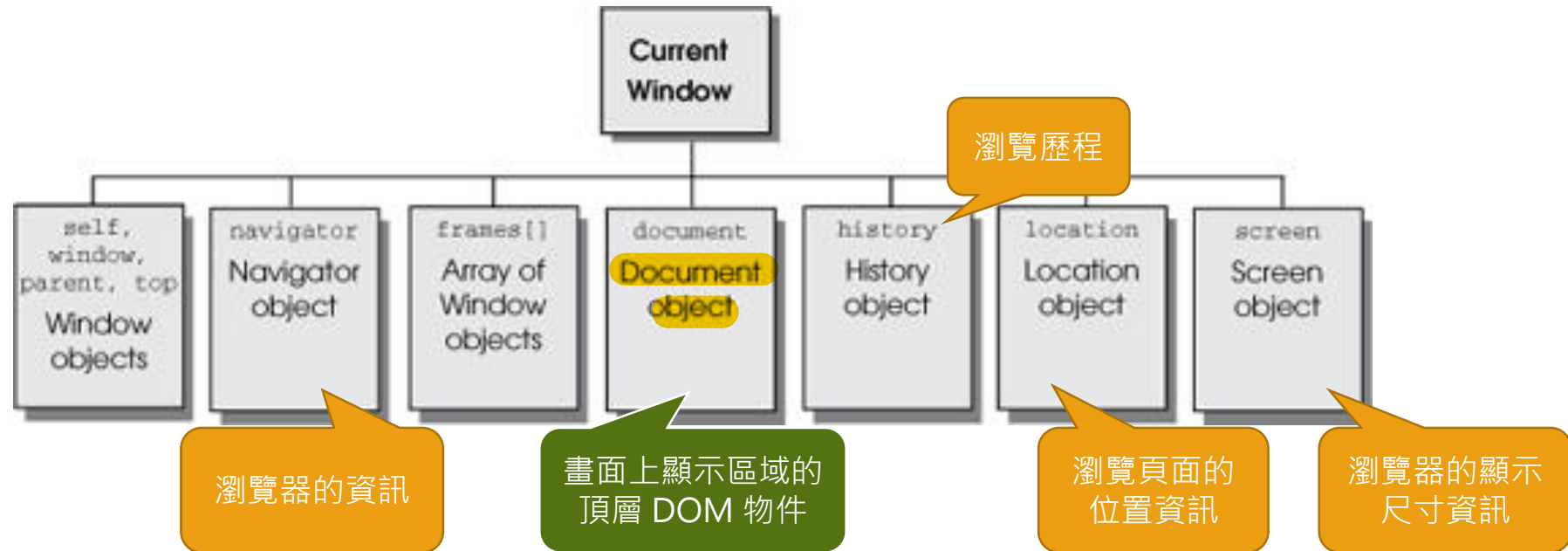


JavaScript 應用

系統開發部 程式設計科

BOM

- BOM (browser object model) 瀏覽器對象模型
- BOM 的核心對象是 window，它表示瀏覽器的一個實例。
- window 處於 JavaScript 結構的最頂層，對於每個打開的視窗，系統都會自動為其定義 window 物件。



HTML DOM

- DOM (document object model) 文件物件模型

- DOM 主要是將 Document 架構為一個樹 (Tree) 的概念。

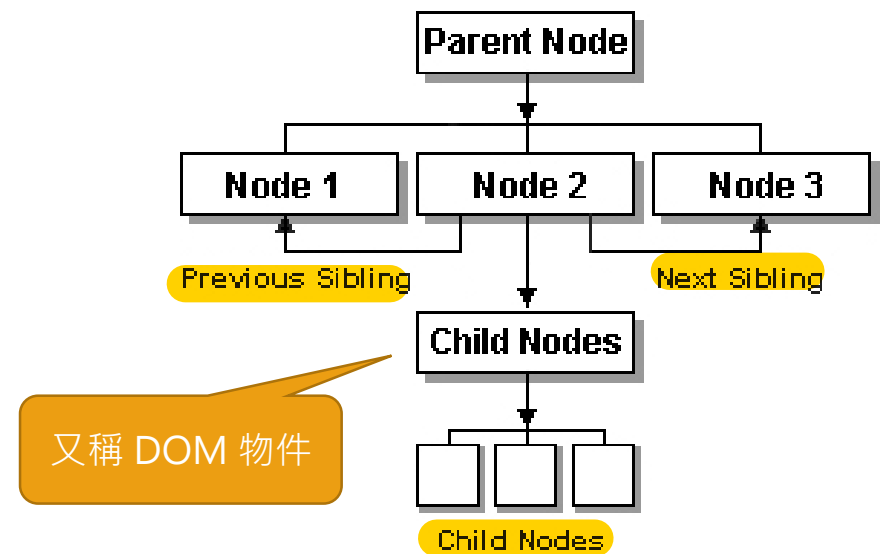
每個 DOM 必須要有一個 document 的根節，

Tree 的組成成份就是節點 (Node) 。

- 當一個 .html 檔被以瀏覽器開啟的時候，瀏覽器就會去解析該檔案的 DOM 。

HTML DOM 把一個個 HTML 元素 (element) 都變成了物件 (object) ，

因此可以透過 JavaScript 設定、存取元素屬性 (attribute) 與內容。



DOM的document其實也是window物件的子物件之一。

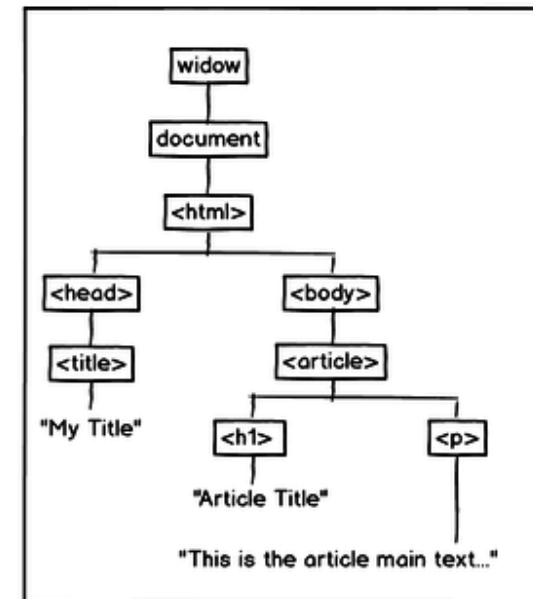
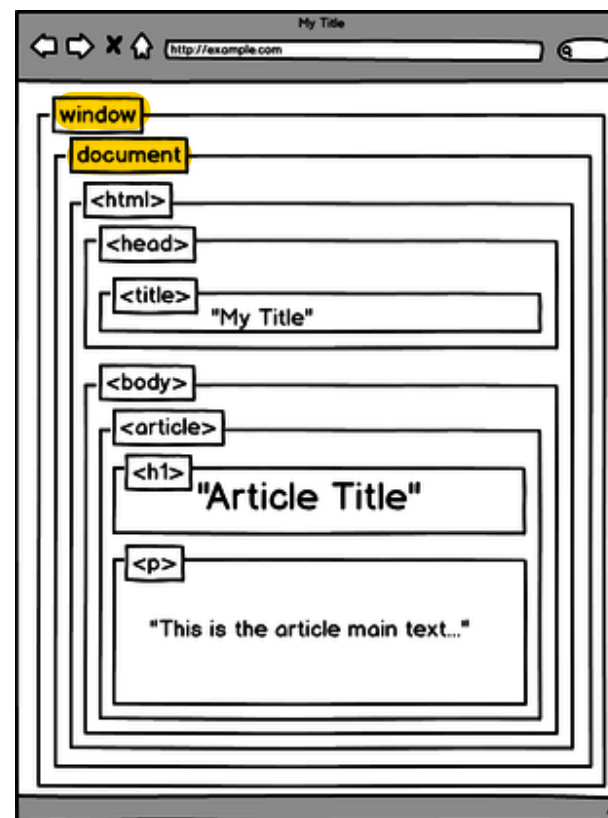
而DOM和BOM最大的區別在於：
BOM JavaScript和『瀏覽器』溝通的窗口，不涉及網頁內容。
DOM JavaScript用來控制『網頁』的節點與內容的標準。

window.document

- 視窗中每個框架都包含一個 window 物件。

- 每個 window 物件都包含一個 document 屬性，

該屬性參考視窗中顯示 html 文件的 Document 物件。



常用 API

DOM document : https://www.w3schools.com/jsref/dom_obj_document.asp

DOM element : https://www.w3schools.com/jsref/dom_obj_all.asp

DOM event : https://www.w3schools.com/jsref/dom_obj_event.asp

The document object is the root node of the HTML document.

The document object is a property of the window object.

The document object is accessed with: window.document or just document

document.getElementById()	onclick (event)
document.createElement()	onchange (event)
element.appendChild()	
element.setAttribute()	element.removeChild()

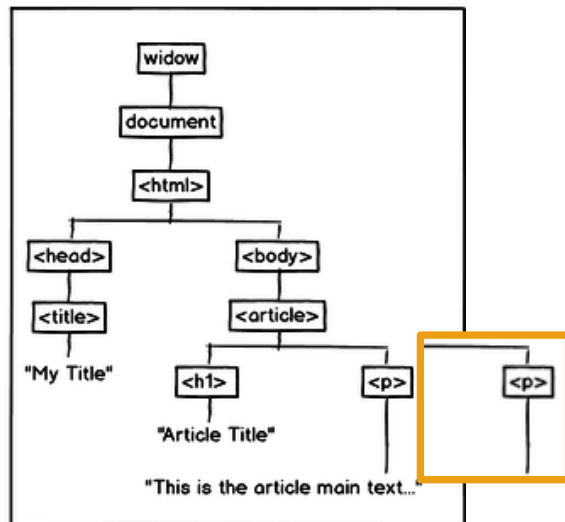
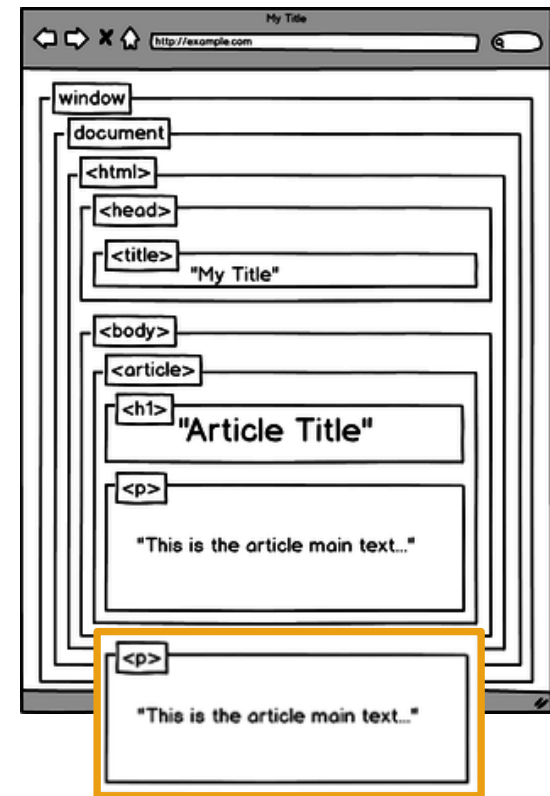
常用 API

- 如何透過 JavaScript 動態找到 DOM

document.getBy...	回傳
getElementById()	Id 是唯一值，只會找到一個， 所以直接回傳 DOM 物件
getElementsByClassName()	其餘都有可能找到多個， 所以把符合的 DOM 物件裝在 [] 回傳 <div>回傳array，所以可以 利用length得到長度來 做迴圈</div>
getElementsByName()	
getElementsByTagName()	

常用 API

- 如何透過 JavaScript 動態多一行文字



- 產生 `<p>`：如果是寫 html

```
<p class="warm"></p>
```

```
var tagP = document.createElement('p');  
tagP.setAttribute('class', 'warm');
```

盡量都用 `.setAttribute`，盡量不要寫成 `class = 'warm'` 這樣(不確定會不會出問題)

- 找到 `<article>`：假設當初 html

```
<article id="show"></article>
```

```
var tagArticle = document.getElementById('show');
```

- 建立關係

```
tagArticle.appendChild(tagP);
```

取值 & 設值

Practice 1

```
<article>
  <p id="title">Practice 1<p>
  <input type="text" id="param" value="A" />
</article>
```

- 透過 JavaScript 動態操作

```
var title = document.getElementById('title');
console.log(title.innerHTML); // Practice 1
title.innerHTML = 'Practice 2';
```

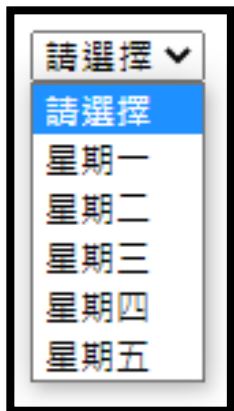
```
var param = document.getElementById('param');
console.log(param.value); // A
param.value = 'B';
```

innerHTML指的是從對象的起始位置到終止位置的全部內容,包括Html標籤。
innerText指的是從起始位置到終止位置的內容,但它去除Html標籤。

Practice 2

js獲取標籤之間的值：
1.透過.innerHTML(建議)
2.透過.innerText

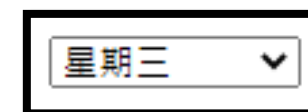
取值 & 設值



```
<select id="day">
  <option value="">請選擇</option>
  <option value="MON">星期一</option>
  <option value="TUE">星期二</option>
  <option value="WED">星期三</option>
  <option value="THU">星期四</option>
  <option value="FRI">星期五</option>
</select>
```

- 透過 JavaScript 動態操作

```
var day = document.getElementById('day');
day.value = 'WED';
console.log(day.value); // WED
```



這邊屬於比較特殊的情況，value也屬於select標籤的屬性

取值 & 設值

- ☒ 星期一
- ☐ 星期二
- ☒ 星期三
- ☐ 星期四
- ☐ 星期五

```
<label><input class="check" type="checkbox" value="MON" checked/><span>星期一</span></label><br>
<label><input class="check" type="checkbox" value="TUE" /><span>星期二</span></label><br>
<label><input class="check" type="checkbox" value="WED" checked/><span>星期三</span></label><br>
<label><input class="check" type="checkbox" value="THU" /><span>星期四</span></label><br>
<label><input class="check" type="checkbox" value="FRI" /><span>星期五</span></label>
```

- 透過 JavaScript 動態操作

```
var checks = document.getElementsByClassName('check');
```

- 取得有勾選的資訊

```
for (var i = 0; i < checks.length; i++) {
    var check = checks[i];
    if(check.checked){
        console.log(check.value); // MON, WED
    }
}
```

- 反向勾選

```
for (var check of checks) {
    check.checked = !check.checked;
}
```

- ☐ 星期一
- ☒ 星期二
- ☐ 星期三
- ☒ 星期四
- ☒ 星期五

Set a event

- `element.onclick = function(){ ... };`

也就是後面寫的會覆蓋前面的

沒有事件疊加的概念，
永遠只會執行最後一次的 function

- `element.addEventListener('click', function(){ ... });`

設定多個事件，每個事件都會執行
(先設定，先執行)

此外，onclick與addEventListener
不會互相影響



Ex.

編譯錯誤： Uncaught TypeError: Cannot set property 'onclick' of **null**

```
<!DOCTYPE html>
<html>
<script>

    document.getElementById('btn_alert').onclick = function(){
        alert('hi');
    };

</script>
<body>
    <button type="button" id="btn_alert">hi</button>
</body>
</html>
```



執行順序

- 瀏覽器解析網頁時，是從頭到尾一段一段解析，

所以程式所在相對的順序會相當重要。

- 由於在執行 JavaScript 程式時，瀏覽器尚未解析到 `<button>` 物件，

所以 `<button>` 尚未產生而回傳 `null`。

- 同理，存取其他 DOM 物件的時候也是會有相同問題。

➤ 解法：將程式碼寫入 `window.onload` 事件中



window.onload

- 觸發時機點：網頁瀏覽器下載完所有網頁檔案時。

(包含：HTML 檔案本身、網頁連結到的圖檔、外部 .css 和 .js 檔)

```
<!DOCTYPE html>
<html>
<head>
<script>
    var init = function(){
        document.getElementById('btn_alert').onclick = function(){
            alert('hi');
        };
    };

    window.onload = init;
</script>
</head>
<body>
    <button type="button" id="btn_alert">hi</button>
</body>
</html>
```

練習 1：practice1.html

- 依照以下指令進行 JS 程式撰寫
 - 按下 btn_1：將 id 為 test 的物件內容換成 "ABC"
 - 按下 btn_2：所有的 input 物件都要新加上一個屬性：style = "color:green;"
 - 按下 btn_3：所有名稱 (name) 為 cost 的物件，如果 value 屬性大於 1000 ，

請將離他最近的上層 TR 物件加上一個屬性：style = "background-color:pink;"

Month cost		income
Jan	<input type="text" value="500"/>	<input type="text" value="4483"/>
ABC	<input type="text" value="11980"/>	<input type="text" value="5631"/>
Mar	<input type="text" value="5689"/>	<input type="text" value="8902"/>
<input type="text" value="ABC"/>	<input type="text" value="green"/>	<input type="text" value="pink"/>

程式小技巧

- 如果已有票號就進行選位等後續動作，否則跳出警示訊息。


```
var action = function(ticketNo){  
    if(ticketNo != ''){  
        // 進行選位等後續動作  
    }else{  
        alert('請先購票');  
    }  
};
```

建議

```
var action = function(ticketNo){  
    if(!ticketNo){  
        alert('請先買票');  
        return;  
    }  
    // 進行選位等後續動作  
};
```

➤ 程式能提早結束就提早結束，可讀性也較好

練習 2 : me.html



英文名：

哪裡人：

正式工作經驗：☐有 ☐沒有

語言：☐國語 ☐台語 ☐其他

如果可以出國旅遊的話，想去~

1	<input type="text"/>
2	<input type="text"/>
3	<input type="text"/>

接續之前成果，編輯 me.html

1. 勾選「其他」要連動後面輸入框是否開放填寫
(取消要清空)
2. 調整「Say HI~」按鈕功能
 - 按下去畫面不重整
 - 先檢核「英文名」為必選
 - 檢核通過，跳出上半部輸入結果的確認視窗
 - 點選「確認」就跳出訊息『Hi~英文名』
 - 檢核失敗，跳出訊息『請填寫英文名』

這個網頁顯示

英文名：AAA
哪裡人：基隆市
正式工作經驗：有
語言：國語 英語

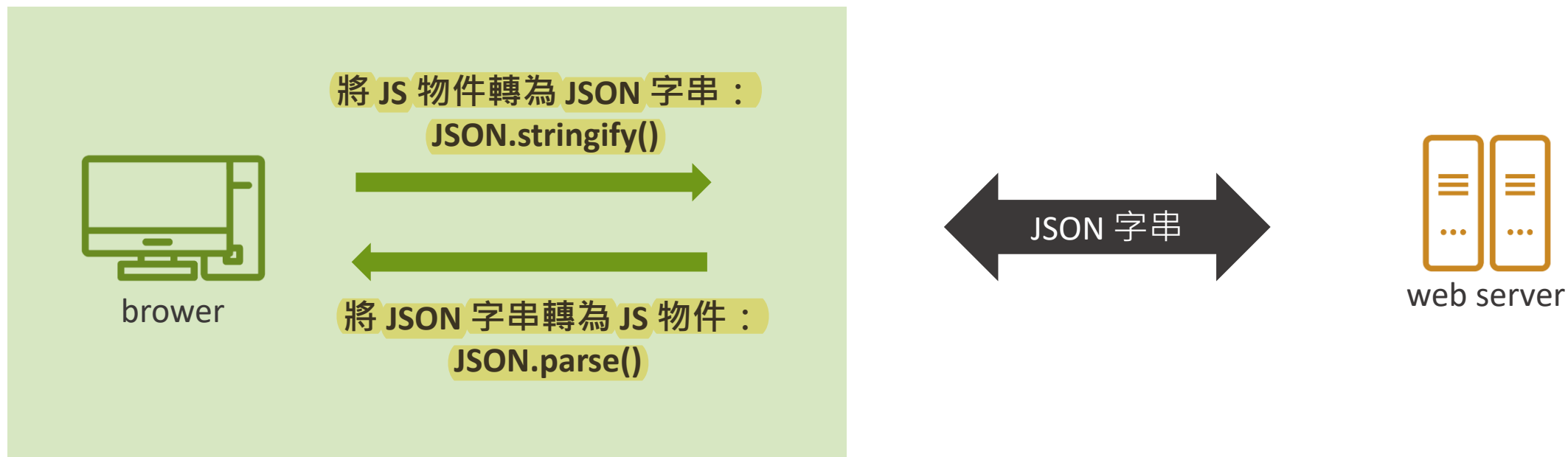
這個網頁顯示

Hi~AAA

- 利用 `confirm()`、`alert()`、換行符號 `\n`
- 考量顯示與背景值的設計

JavaScript Object Notation

- JSON 是受到 JavaScript 物件實字 {} 語法啟發的傳輸格式，是一種輕量級資料交換格式



JavaScript Object Notation

- JSON 支援格式：String、number、Boolean、Array、Object、null

```
var data = {name:'AAA', age:30, work:null, sex:undefined, isMarr:false, lottory:[1,2]};
```

```
var dataJSON = JSON.stringify(data)
```

```
'{"name":"AAA","age":30,"work":null,"isMarr":false,"lottory":[1,2]}'
```

```
var dataObj = JSON.parse(dataJSON);
```

```
{name:'AAA', age:30, work:null, isMarr:false, lottory:[1,2]};
```

練習 3：practice3.html

縣市：

行政區：

編輯 practice3.html

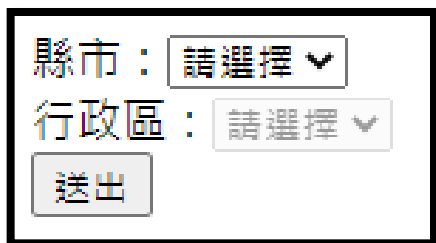
- 用 JS 解析 backJson 字串
 - 取得所有縣市放入縣市下拉選單
 - 取得所有行政區放入行政區下拉選單 (不分縣市)

```
<option value="台北市">台北市</option>
```

```
<option value="001">中正區</option>
```

```
backJson = {  
  datas:[  
    {  
      districts: [{zip:100, name:'中正區'}]  
      ,name: '台北市'  
    }  
  ]  
}
```

練習 4：practice4.html



縣市：請選擇 ▼
行政區：請選擇 ▼
送出

接續練習 3：複製 practice3.html 另存 practice4.html

- 解析 backJson 字串，進行資料整理
 - 縣市選單**改變時**連動行政區選單
 - 縣市沒選：行政區只有請選擇並不開放點選
 - 縣市有選：僅將該縣市的行政區放入下拉選單
- 新增「送出」按鈕：
 - 檢核行政區為必選
 - 檢核失敗，跳出『請選擇行政區』
 - 檢核通過，跳出該行政區代碼

jQuery

jQuery

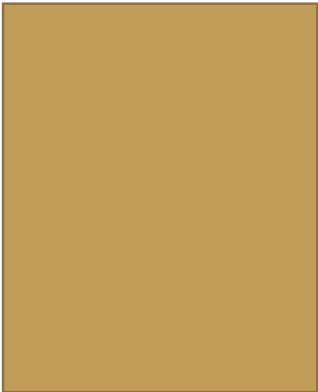
- jQuery 是一套跨瀏覽器的JavaScript函式庫，簡化 HTML 與 JavaScript 之間的操作。
(簡化了在 HTML 文件裡面尋找 DOM 物件、處理事件、製作動態效果等)
- 使用時須引用：`<script type="text/javascript" src="jquery.js"></script>`。

JavaScript	jQuery
<pre>span.innerHTML = 'test'; input.value = 'test'; document.createElement('tr'); button.onclick = function(){...};</pre>	<pre>span.html('test'); input.val('test'); \$('<tr/>'); button.click(function(){...});</pre>

jQuery

- 注意引用順序，須先於其他使用 jQuery 撰寫的 js 檔

jQuery.js



my.js



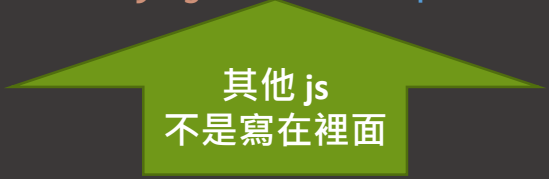
demo.html

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="my.js"></script>
<script>

    window.onload = function(){

    };

</script>
```



- 如果先引用 my.js，瀏覽器會不認得 \$ 而編譯錯誤

css selector

- 選擇器：`$('...')`
- 常用選擇器
 - ID 選擇器：`$('#id')`
 - 類型選擇器：`$('tag')`
 - Class 選擇器：`$('.class')`
 - 屬性選擇器：`$('[attr=attrVal]')`
- 取回的元素型態 (`type`) 是什麼？
 - ID 選擇器：返回一個 jQuery 物件
 - 其餘選擇器：返回一個 jQuery 陣列 (`array`) 型態物件，陣列裡為 DOM 物件。

這邊容易搞錯喔，注意！

所以此時如果想抓第0個以及第1個指標中的物件使用[0], [1]這種取法取到的會是DOM物件，如果為了方便操作想取的是jQuery物件可以前面加個錢幣符號或是改用.eq(0), .eq(1)這種取法解決。



DOM 物件 v.s. jQuery 物件

```
//<input type="text" id="test" value="My input"/>

var domObj = document.getElementById('test');
var jqueryObj = $('#test');

console.log(domObj.value); // My input
console.log(jqueryObj.val()); // My input

//jQuery 物件只能使用 jQuery 的方法；而 DOM 物件只能使用 DOM 物件的方法。
console.log(domObj == jqueryObj); // false
console.log(domObj.val()); // domObj.val is not a function
console.log(jqueryObj.value); // undefined
```

```
//DOM 轉為 jQuery 物件：只要將 DOM 傳入 $()
domObj = $(domObj);
console.log(domObj.val()); // My input

//用 get() 方法將 jQuery object 轉換為 JavaScript DOM 元素集合 => Array
jqueryObj = jqueryObj.get(0);
console.log(jqueryObj.value); // My input
```

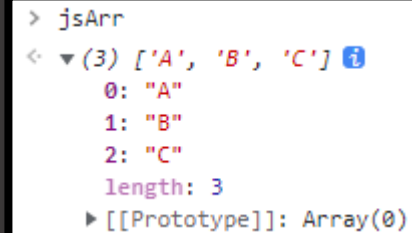
jQuery

- jQuery 方法執行完會回傳呼叫它的 jQuery 物件，所以對同一元素做操作可以一直 . 下去

```
$('#myInput').val('AAA').attr('style', 'color:green;').addClass('myClass');
```

- JavaScript 陣列 v.s. jQuery 陣列

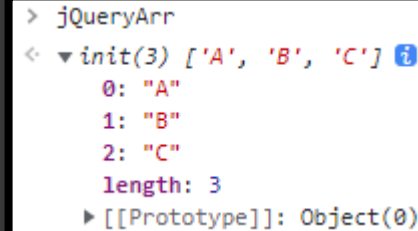
```
var jsArr = ['A', 'B', 'C'];
```



```
> jsArr
< ▼ (3) ['A', 'B', 'C'] ⓘ
  0: "A"
  1: "B"
  2: "C"
  length: 3
  ► [[Prototype]]: Array(0)
```

```
for(var jsEle of jsArr){
  console.log(jsEle);
}
```

```
var jqueryArr = $(['A', 'B', 'C']);
```



```
> jqueryArr
< ▼ init(3) ['A', 'B', 'C'] ⓘ
  0: "A"
  1: "B"
  2: "C"
  length: 3
  ► [[Prototype]]: Object(0)
```

```
jqueryArr.each(function(ind, ele){
  console.log(ele);
});
```

練習 5：practice1.html

- 將 practice1.html 原先 JS 程式用 jQuery 改寫
 - 按下 btn_1：將 id 為 test 的物件內容換成 "ABC"
 - 按下 btn_2：所有的 input 物件都要新加上一個屬性：style = "color:green;"
 - 按下 btn_3：所有名稱 (name) 為 cost 的物件，如果 value 屬性大於 1000 ，

請將離他最近的上層 TR 物件加上一個屬性：style = "background-color:pink;"

Month cost		income
Jan	<input type="text" value="500"/>	<input type="text" value="4483"/>
ABC	<input type="text" value="11980"/>	<input type="text" value="5631"/>
Mar	<input type="text" value="5689"/>	<input type="text" value="8902"/>
<input type="text" value="ABC"/>	<input type="text" value="green"/>	<input type="text" value="pink"/>

Thank You
