

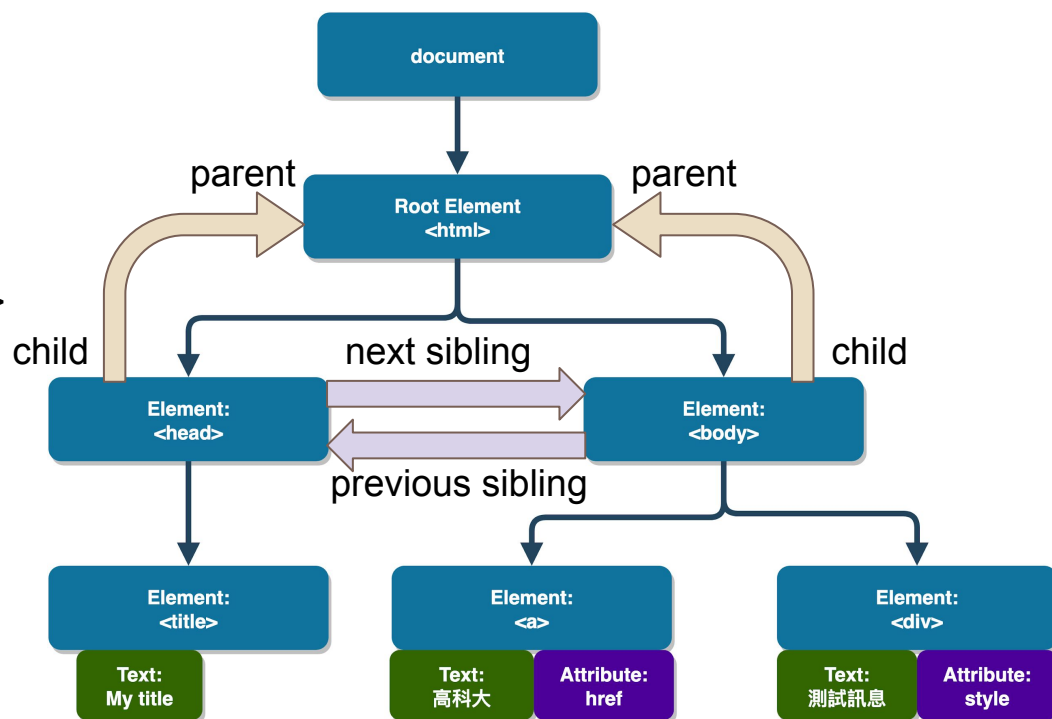
DOM



何謂DOM

- ❖ Document Object Model 文件物件模型
- ❖ 將HTML文件的各個Tag(文字、圖片等等)都定義成物件，並最終形成一個樹狀結構

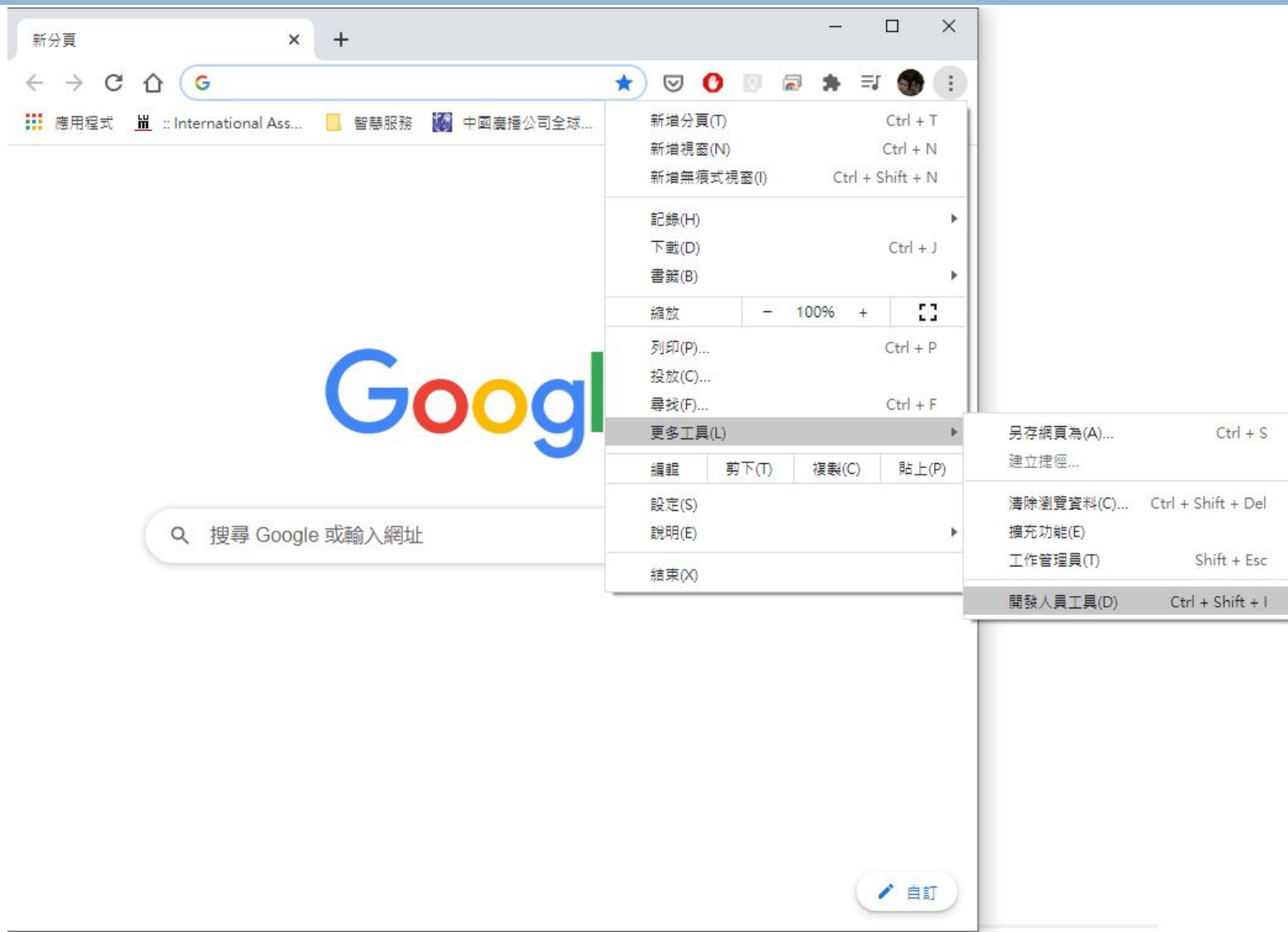
```
<html>  
<head>  
  <title>My title</title>  
</head>  
<body>  
  <a href="http://www.nkust.edu.tw">高科大</a>  
  <div style="color: black;">測試訊息</div>  
</body>  
</html>
```



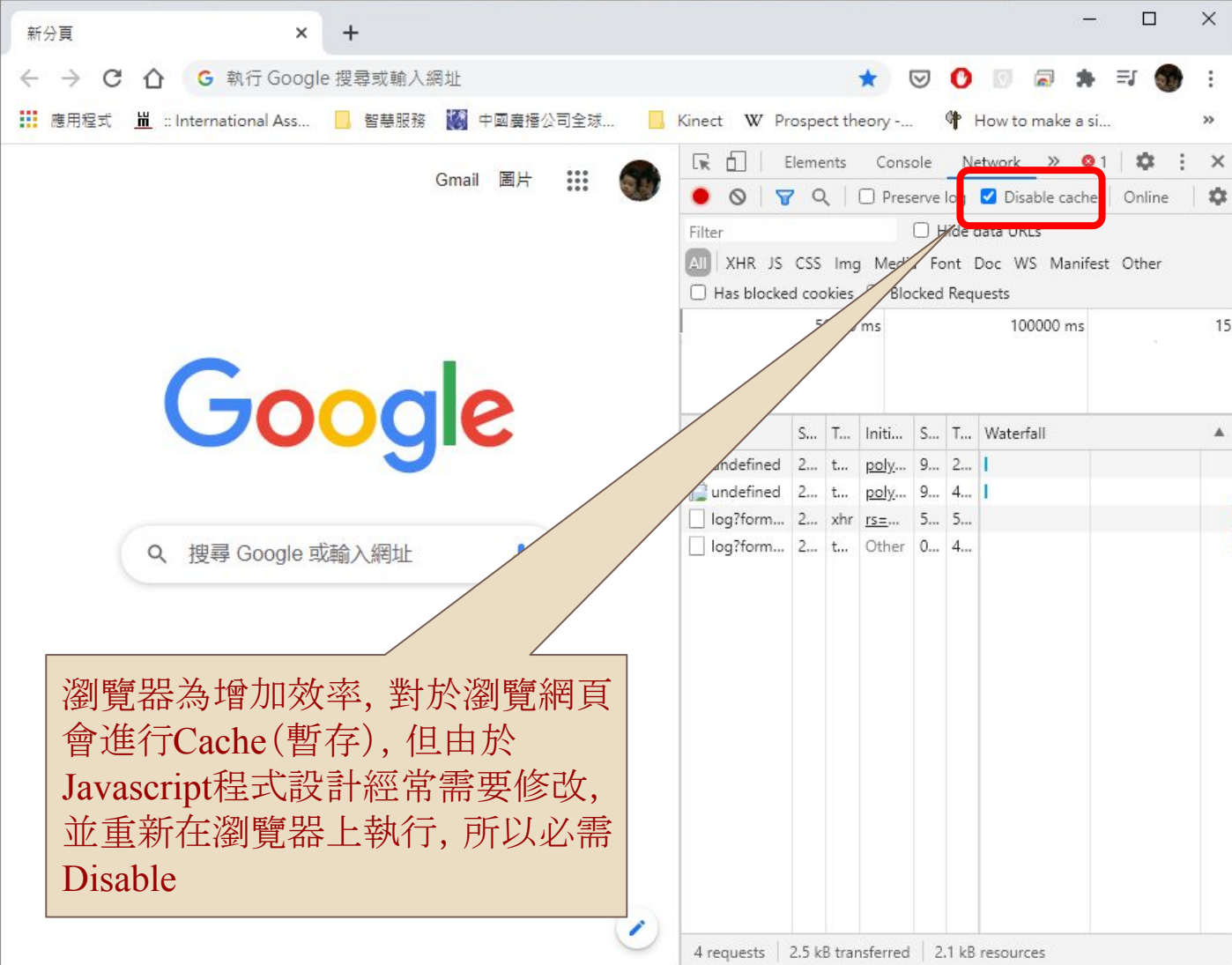
DOM操作

- ❖ 取得符合id名稱為‘**idName**’的element
 - `document.getElementById(‘idName’)`
- ❖ 取得符合tag名稱為‘**tagName**’的所有element集合
 - `document.getElementsByTagName(‘tagName’)`
- ❖ 取得符合class名稱為‘**className**’的所有element集合
 - `document.getElementsByClassName(‘className’)`
- ❖
- ❖ 目前DOM操作, 均大多透過jQuery完成

Chrome 開發人員工具



Network -> Disable Cache



The screenshot shows a web browser window with the Google homepage. The Network panel is open on the right side, and the 'Disable cache' checkbox is checked and highlighted with a red box. A callout box points to this checkbox with the following text:

瀏覽器為增加效率，對於瀏覽網頁會進行Cache(暫存)，但由於Javascript程式設計經常需要修改，並重新在瀏覽器上執行，所以必需Disable

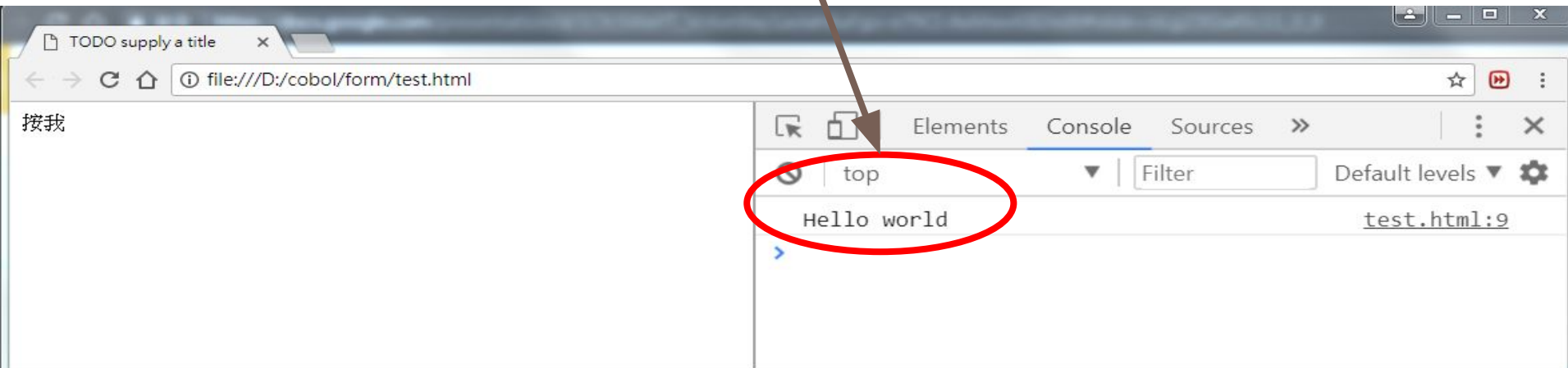
	S...	T...	Initi...	S...	T...	Waterfall
undefined	2...	t...	poly...	9...	2...	
undefined	2...	t...	poly...	9...	4...	
log?form...	2...	xhr	rs=...	5...	5...	
log?form...	2...	t...	Other	0...	4...	

4 requests | 2.5 kB transferred | 2.1 kB resources

HTML 元素與事件呼叫—console.log()

```
<html>
  <head>
    <script>
      function hello(){
        console.log("Hello world!");
      }
    </script>
  </head>
  <body>
    <div onclick='hello()'>按我</div>
  </body>
</html>
```

很適合用在debug



jQuery



jQuery

- ❖ 最早由John Resig於2006年1月的BarCamp NYC上發佈第一個版本
- ❖ 一套物件導向式簡潔輕量級的 JavaScript Library
- ❖ 可以用最精簡的程式碼達到跨瀏覽器 DOM 操作、事件處理、設計頁面元素動態效果、AJAX 互動等。
- ❖ 為何使用jQuery
 - 官方文件詳細 (重要)
 - 壓縮後大約 88kB (v3.5.1)
 - 解決跨瀏覽器 (cross-browser) 的問題
 - 支援 CSS3 選擇器
 - 社群活躍, 有很多免費的程式庫
- ❖ 版本介紹
 - 1.x: 兼容IE678, 官方只做BUG維護, 功能不再新增
 - 2.x: 不兼容IE678, 官方只做BUG維護, 功能不再新增
 - 3.x: 不兼容IE678, 只支持最新的瀏覽器; 很多老的jQuery插件不支持3.x版。目前該版本是官方主要更新維護的版本。

開始使用

- ❖ 自行下載jQuery
 - 下載 jQuery Library <http://jquery.com/download/>
 - 將下載後的 JS 檔放進 `<head>` 及 `</head>` 之間
`<script src="jquery-3.5.1.min.js"></script>`
- ❖ 透過 CDN (Content Delivery Network), 在 `<head>` 及 `</head>` 之間放入下列程式碼
 - Google CDN
`<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>`

jQuery 語法

❖ 基本語法: `$(selector).action()`

- `$` 是用來定義存取 jQuery
- `(selector)` 用來尋找 HTML elements
- `action()` 對上述 HTML element(s) 的動作
- 若出現 **\$ not found** 表示 jquery 載入失敗

❖ Examples:

- `$(this).hide()` - 隱藏目前的 element
- `$("p").show()` - 顯示所有 `<p>` elements
- `$(".test").hide()` - 隱藏所有 `class="test"` 的 elements
- `$("#test").show()` - 顯示 `id="test"` 的 element

Web Event語法 -1

❖ 直接在element後面加上{on事件}函數：

➤ `<div onclick=test()>測試</div>`

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
  function test(obj){
    obj.hide();
  }
</script>
</head>
<body>
  <p>If you click on me, I will disappear.</p>
  <p onclick="test($(this))">按我就消失</p>
  <p onclick="test($(this))">按我也會消失</p>
</body>
</html>
```

Web Event語法 -2

❖ jQuery常用寫法：

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
    $(document).ready(function(){
        $("p").click(function(){
            $(this).hide();
        });
    });
</script>
</head>
<body>
    <p>If you click on me, I will disappear.</p>
    <p>按我就消失</p>
    <p>按我也會消失</p>
</body>
</html>
```

安裝jQuery Code Snippets

The screenshot shows the Visual Studio Code interface with the Extensions Marketplace open. The search bar at the top left contains the text "jquery". The search results list several extensions, with "jQuery Code Snippets" by Don Jayamanne at the top. A red circle with the number "1" is around the search bar, a red circle with the number "2" is around the extension name, and a red circle with the number "3" is around the "Install" button. The main panel shows the details for the "jQuery Code Snippets" extension, including its logo, author (Don Jayamanne), version (v0.0.1), and a description: "Over 130 jQuery Code Snippets for JavaScript code." Below the description, it says "Just type the letters 'jq' to get a list of all available jQuery Code Snippets." An inset window shows the "index.js" file in the "vscode-jscs" project, where the text "jq" is typed, and a list of jQuery methods is displayed as a dropdown menu, with "jqAttrSetFn" selected.

Visual Studio Code interface showing the installation of the **jQuery Code Snippets** extension.

Left Panel (Extensions Marketplace):

- Search: **jquery**
- Results:
 - jQuery Code Snippets** 0.0.1 (Don Jayamanne) **Install**
 - jQuery Snippets** 1.0.0 (Hridoy) **Install**
 - no jQuery Snippets** 0.1.3 (Nkzq) **Install**
 - A-super-comprehensive** 1.9.8 (代码补全合集(vue-webpack-bootstrap... 雪导) **Install**
 - quickstart** 0.1.0 (HTML5 snippet for jQuery 3, Bootstra... lolkush) **Install**
 - mdbsnippets** 1.4.7 (MDBootstrap snippet generator Mdbootstrap) **Install**
 - EventListener Intellisense** 0.0.1 (Intellisense for JS Event Listeners Bernardo Baldiviezo) **Install**
 - Stickode** 0.0.1 (Cloud-based snippet, Source sharing ... Stickode) **Install**
 - A-snippets-for-gsap** 1.0.3 (GSAP动效引擎 雪导) **Install**
 - VSC-Kendo-Components** 0.2.0

Right Panel (Extension Details):

- jQuery Code Snippets** (donjayamanne.jquerysnippets)
- Don Jayamanne | 880,203 | ★★★★★ | v0.0.1
- Over 130 jQuery Code Snippets
- Install**
- Details | Feature Contributions
- jQuery Code Snippets**
- Over 130 jQuery Code Snippets for JavaScript code.
- Just type the letters 'jq' to get a list of all available jQuery Code Snippets.

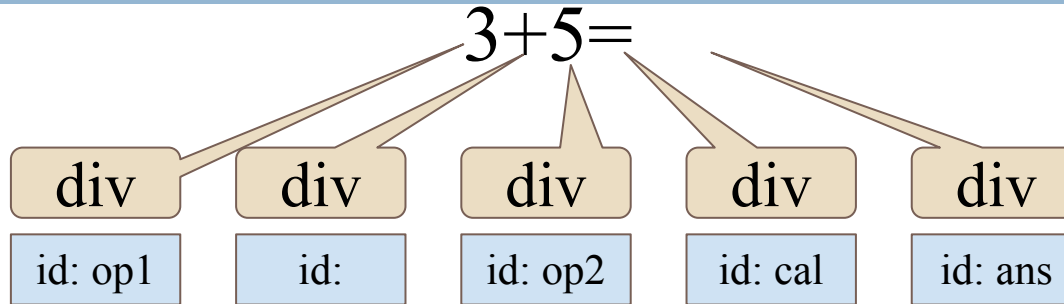
Inset Window (Extension Development Host):

- File Edit View Goto Help
- index.js test
- 1 | jq
- Dropdown list of jQuery methods:
 - jqAppendTo
 - jqAttrGet
 - jqAttrRemove
 - jqAttrSet
 - jqAttrSetFn** (Set one or more attributes for the set of matched elements.)
 - jqAttrSetObj
 - jqBefore
 - jqBind
 - jqBindWithData

jQuery — 設定或取得相關內容

- ❖ `text()`: 設定或取得 elements 的純文字內容
- ❖ `html()`: 設定或取得 elements 的HTML內容
- ❖ `val()`: 設定或取得表單元件的value, 例如: 文字框等
- ❖ `attr()`: 取得element的屬性值
- ❖ `css()`: 設定或取得css的value
- ❖ 所有取得內容均為字串型態, 但設定內容時, 允許各種型態
- ❖ 字串型態⇒整數型態: `parseInt()`; 字串型態⇒浮點數型態: `parseFloat()`

練習



- ❖ 步驟1: 在<body>作畫面

```
<div id="op1">3</div>+<div id="op2">5</div><div id="do">=</div><div id="ans"></div>
```

- ❖ 步驟2: 在<head>載入jquery

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

- ❖ 步驟3: 在<head>設定需要的CSS樣式

```
<style>
  div {
    display: inline;
  }
</style>
```

- ❖ 步驟4: 在<head>撰寫需要的javascript code

```
<script>
  $(document).ready(function(){
    $("#do").click(function(){
      let op1 = $("#op1").text();
      let op2 = $("#op2").text();
      let ans = op1 + op2;
      $("#ans").text(ans);
    });
  });
</script>
```

錯誤



範例一 設定CSS

按我變藍色 按我變紅色

id: blue

id: red

- ❖ 步驟1: 在<body>作畫面
- ❖ 步驟2: 在<head>載入jquery
- ❖ 步驟3: 在<head>設定需要的CSS樣式
- ❖ 步驟4: 在<head>撰寫需要的javascript code

```
<script>
    $(document).ready(function(){
        $("#blue").click(function(){
            $("#blue").css("color", "blue");
        });
        $("#red").click(function(){
            $("#red").css("color", "red");
        });
    });
</script>
```


建立輸入欄位

17

● 單列文字

王小明

```
<input type="text" id="username">
```

● 密碼

.....

```
<input type="password" id="passwd">
```

● 多列文字

```
<textarea rows="2" cols="20" id="comment">
```

PHP的全名為Hypertext Preprocessor,
是個被廣泛運用在網頁程式撰寫的語言。

```
</textarea>
```

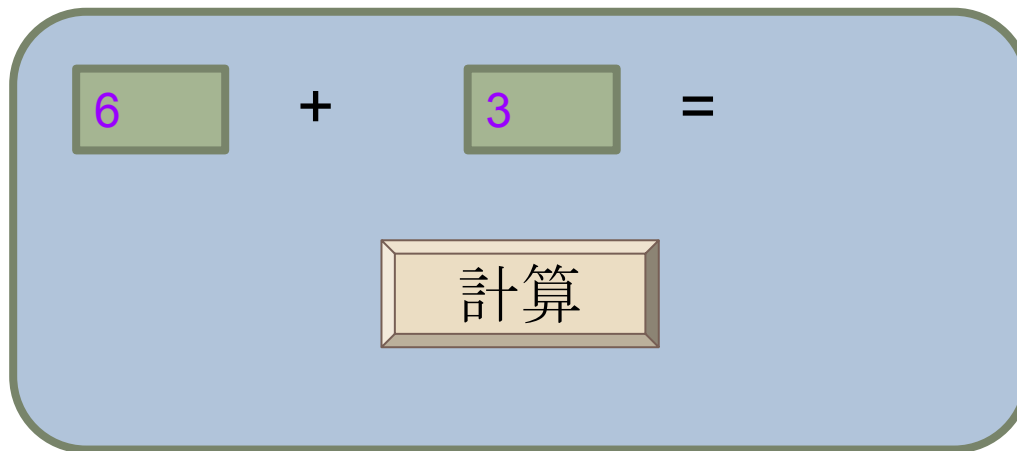
PHP全名為的Hypertext
Preprocessor, 是個被..

● 按鈕

```
<input type="button" value="送出" id="btn">
```

送出

練習一 加法計算機



提示：

1. 作畫面
2. 設定事件(click)
3. 撰寫事件處理程序

下拉式選單

<select id="DropDown">

<option value="選項值1">說明文字1</option>

<option value="選項值2">說明文字2</option>

...

</select>

- **說明文字**: 顯示在網頁上的文字
- **選項值**: 實際傳回的表單值
 - ex: 若選擇項目二, 傳回的表單值就是 "DropDown = 選項值 2"。
 - value 屬性並非必要屬性, 如果沒有則 "DropDown = 說明文字 2"。
- 若某個 <option> 加入 selected, 該項會預設勾選狀態
- **size="n"**: 可建立多行的列表欄
- **multiple 屬性**: 讓選單可一次選擇多個項目

下拉式選單

單選

```
<select id="learning">
  <option value="1">JSP</option>
  <option value="2" selected>PHP</option>
  <option>ASP</option>
  <option value="4">C++</option>
  <option value="5">Perl</option>
</select>
```

<!-- 送出值:1 -->
<!-- 預設值 -->
<!-- 送出值:ASP -->
<!-- 送出值:4 -->
<!-- 送出值:5 -->

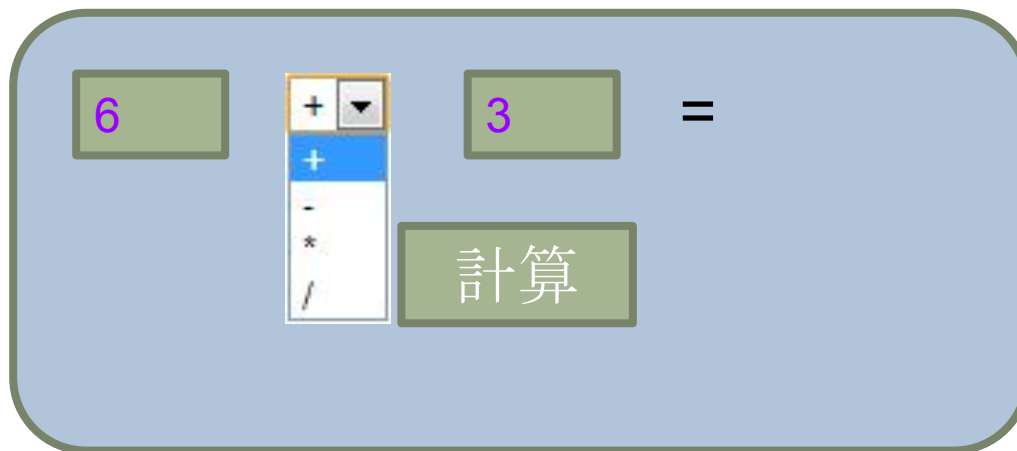
多選

```
<select id="learning" multiple>
  <option value="1">JSP</option>
  <option value="2" selected>PHP</option>
  <option>ASP</option>
  <option value="4">C++</option>
  <option value="5">Perl</option>
</select>
```

<!-- 送出值:1 -->
<!-- 預設值 -->
<!-- 送出值:ASP -->
<!-- 送出值:4 -->
<!-- 送出值:5 -->

Q: 選單允許多選, 會以相同名稱送出多筆資料, PHP 應該如何讀取資料?

練習一 加減計算機



相加: 9
相減: 3
相乘: 18
相除: 2

練習一 學生成績

國文

英文

數學

平均

計算

及格用藍色顯示
不及格用紅色顯示

Web Event

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	upload

練習：將上述3個例子改用keyup事件再作一次

建立輸入欄位

24

● 單選鈕：☉男 ☉女

`<input type="radio" name="gender" id="gender" value='male'/>` 男

`<input type="radio" name="gender" id="gender" value='female'/>` 女

● 多選鈕：☑電影 ☐電腦 ☑讀書

`<input type="checkbox" id="computer" value="cp"/>`電腦

`<input type="checkbox" id="movie" value="mv"/>` 電影

一定要設定 name, 否則無法顯現radio的效果

● 取得值的方式較為特別：

- `$("input[id=gender]:checked").val()`
- `$("input[name=gender]:checked").val()`

練習一 學生成績

國文

英文

數學

平均

特殊生: ☒是 ☐否

及格用藍色顯示
不及格用紅色顯示

特殊生平均加10分

一般程式語言範圍與內容

26

- 變數
 - 型態: 數值、字串、布林
- 運算
 - 算數運算: +, -, *, /, %
 - 比較運算: >, <, ==, >=, <=, !=
 - 邏輯運算: &&, ||, !
- 選擇結構
 - if ... else
 - switch ... case
- 重覆結構
 - for
 - while
 - foreach
- 陣列
- 副程式/函數

變數

27

❖ 無型態宣告，其型態依據變數值決定

❖ 變數宣告

- ES6後，請使用let，**不要再用var**
- 先前未宣告之變數與var類似，但在特殊情況下，容易出錯
- let a;
- let a = 5;

❖ 常數宣告

- **常數只給一次值之後不得改變，經常使用**
- const op = \$("#op").val();
- const PI = 3.14159;

迴圈—for

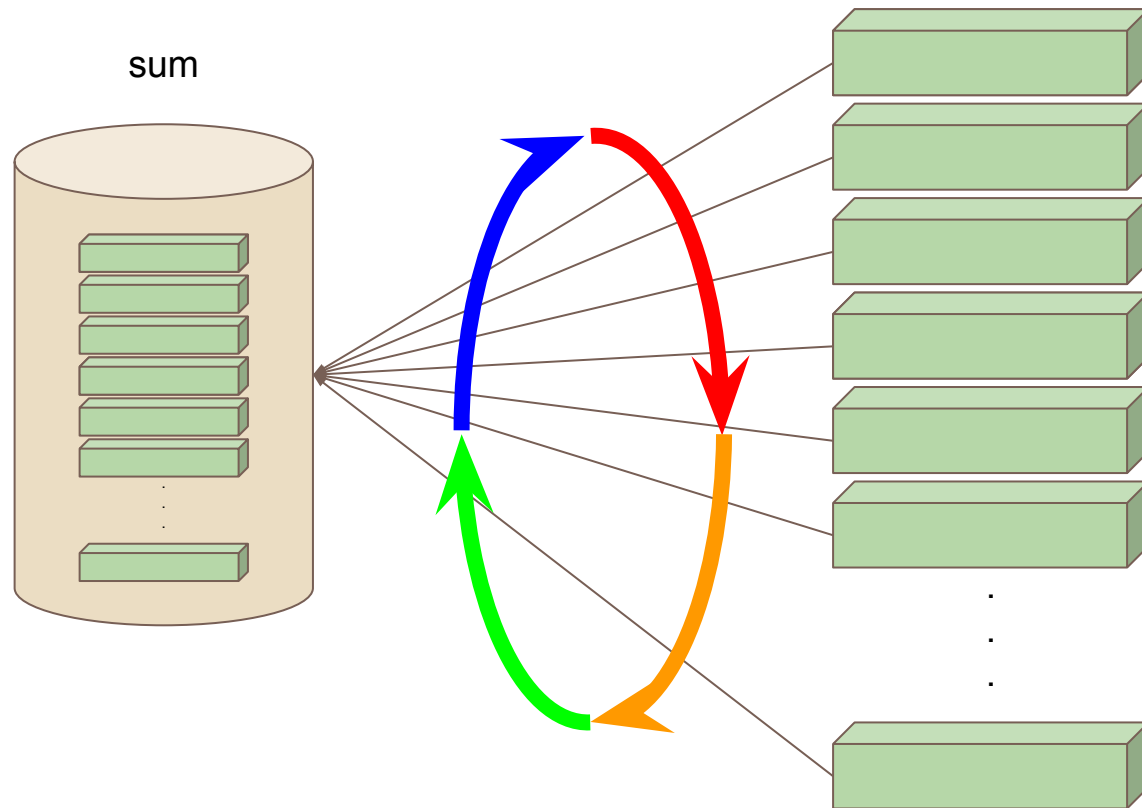
28

- 語法：

```
for(初始運算式; 條件運算式; 控制運算式){  
    //迴圈中要重覆執行的程式碼  
}
```

- **初始運算式** 只有第 1 次進入 for 迴圈之前會被執行
- **條件運算式** 則和 while 一樣, 是用以判斷是否要繼續執行大括號的內容
- **控制運算式** 則是每一輪迴圈執行結束後都會執行的
- 練習：輸入兩數a, b, 計算 $a+...+b=?$
- 練習：輸入兩數a, b, 計算 $\int_a^b (x^2 + 1) dx$

累加



迴圈—while

30

- 語法：

```
while(條件運算式){  
    //迴圈中要重覆執行的程式碼  
}
```

- 當條件運算式為 `true` 時, 就會重複執行大括號中的程式碼。
- 練習: 輸入兩數, 計算最大公因數

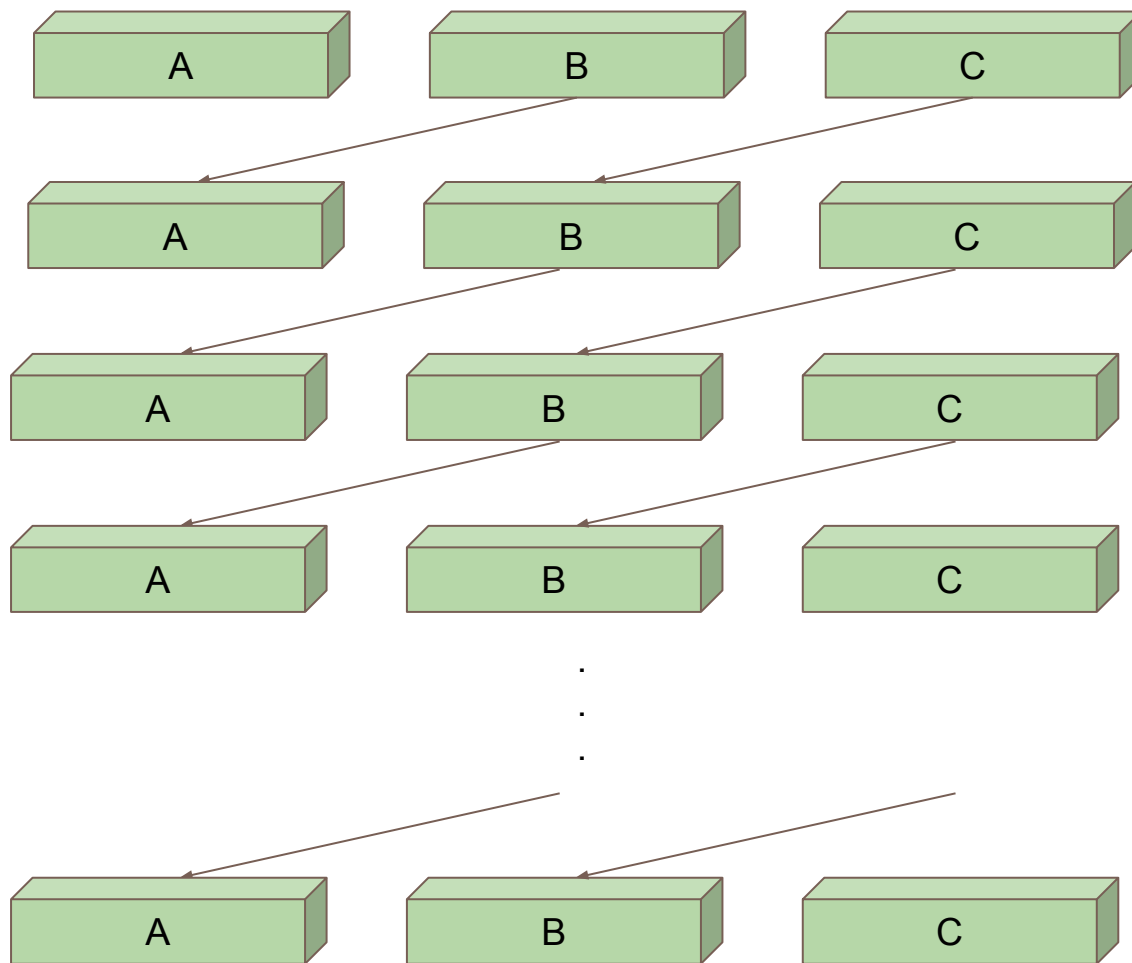
迴圈—do while

31

- 與while類似，但適用於迴圈『至少要執行一次』的情況，其語法如下。

```
do {  
    //迴圈中要重覆執行的程式碼  
} while(條件運算式);
```

疊代

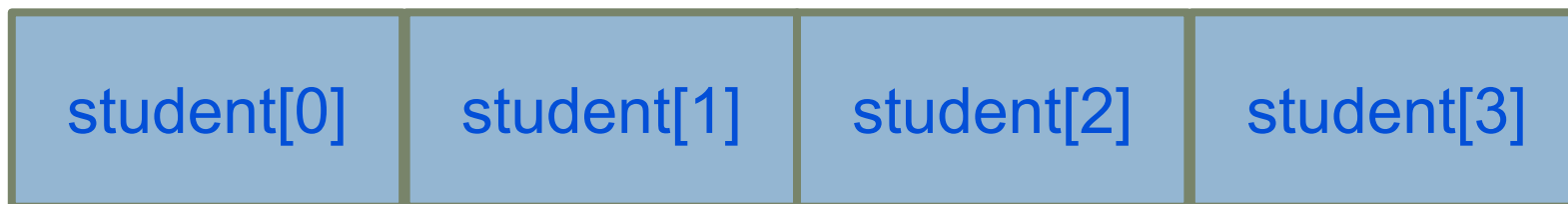


陣列

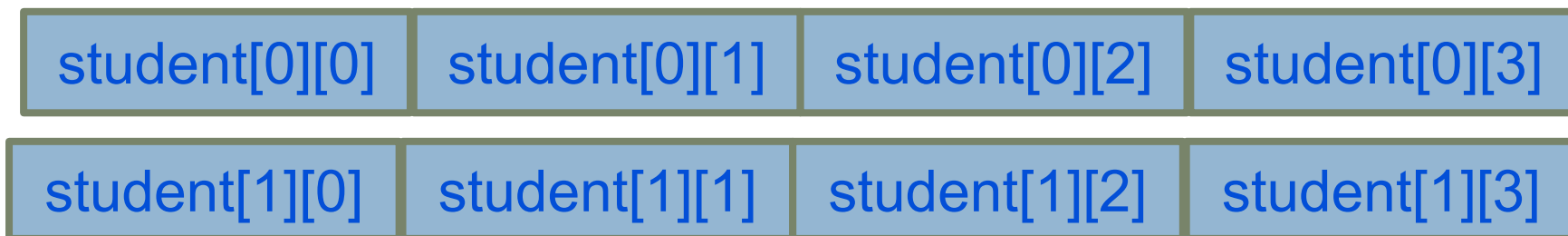
- 程式設計的功能之一，處理大量的資料
- 可用單一變數代表一群(相同型態)變數集合，屬(循序型/物件型)資料結構。
- 這些資料可經過適當排列(一維陣列，二維陣列等)，方便後續程式撰寫
- 陣列存取以『索引』值(index)為指標，且索引值必須是整數或字串。
- 透過索引可存取陣列個別的元素，索引必需存在於陣列的範圍。

陣列

student 一維陣列



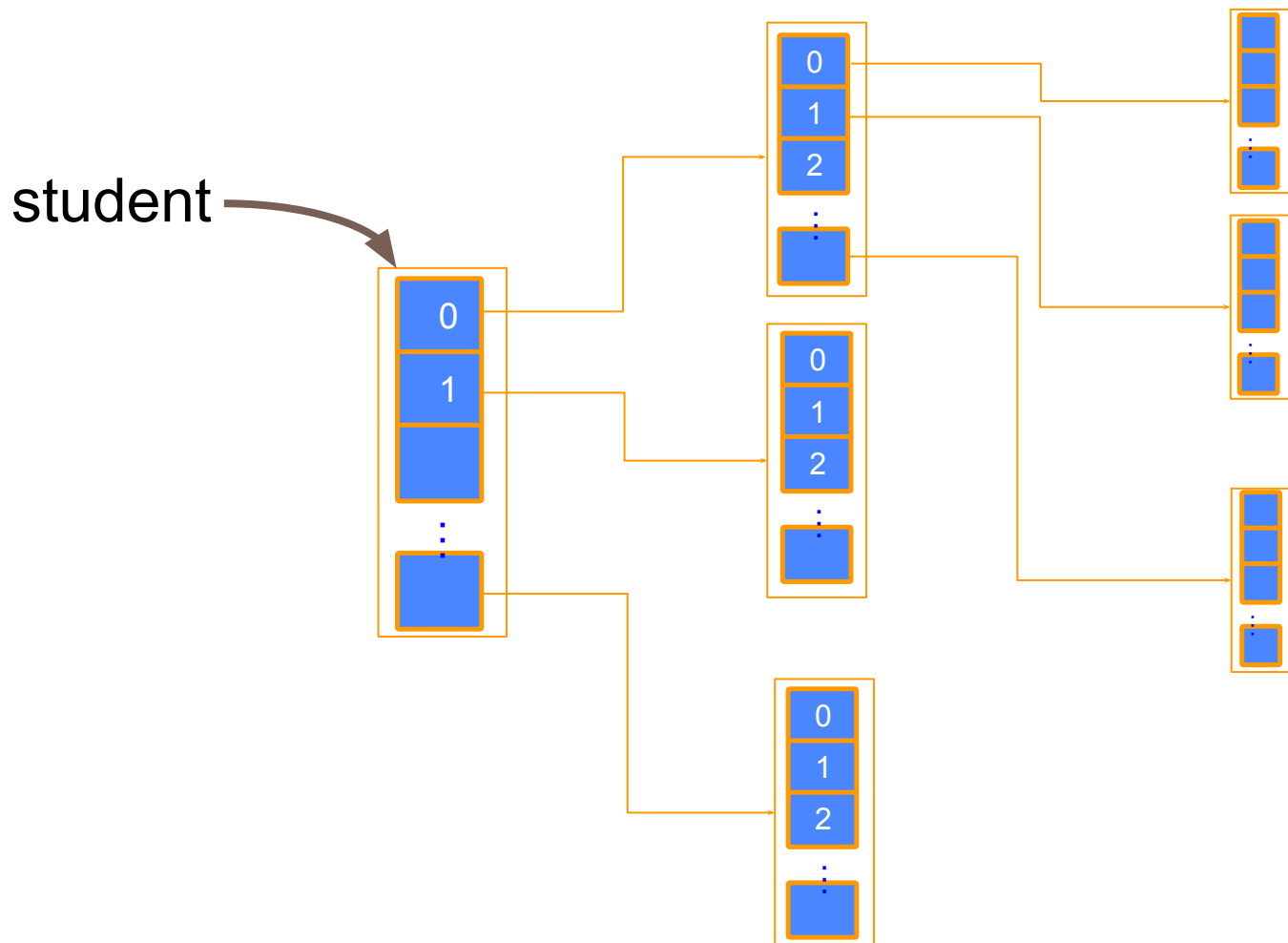
student 二維陣列



二維陣列與多維陣列

35

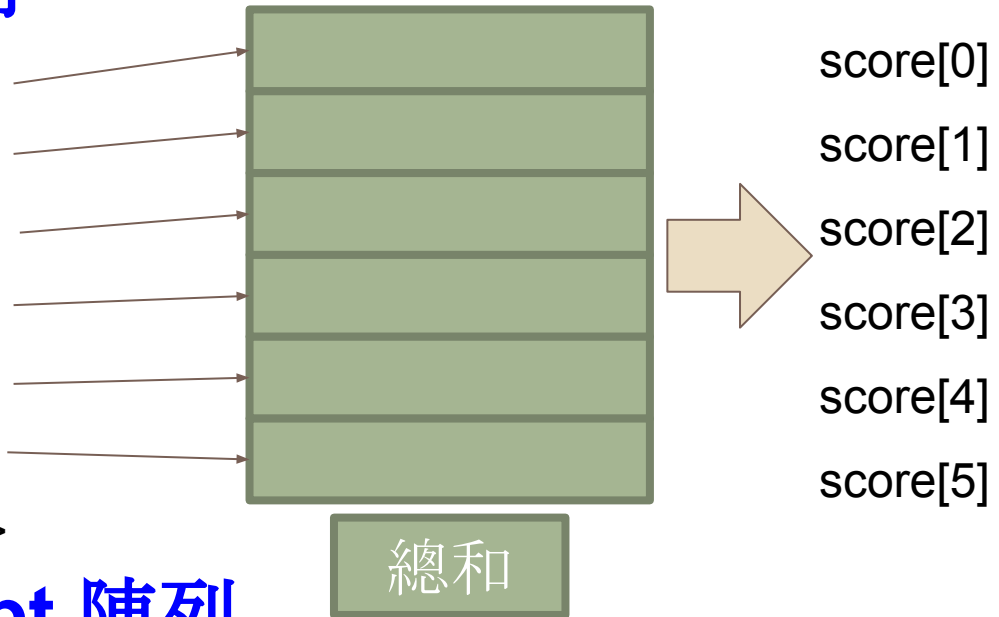
- 二維以上陣列其實可以看成多個一維陣列的組合。



表單元件陣列

HTML 製作元件陣列

```
<input type="text" id="score"><br>  
<input type="text" id="score"><br>  
<input type="text" id="score"><br>  
<input type="text" id="score"><br>  
<input type="text" id="score"><br>  
<input type="text" id="score"><br>  
<input type="button" value="送出">
```



表單陣列=>Javascript 陣列

```
let score = new Array();  
$("input[id=score]").each(function(index, element){  
    score[index] = $(element).val();  
});  
score.push($(element).val());  
score.push($(this).val());
```

練習

Ex1: 輸入全班成績，計算全班平均？

Ex2: 承上題，若全班平均及格則以藍色顯示，否則以紅色顯示。

編號	總分
1	
2	
...	...
50	
平均	

迴圈—forEach

38

- 當需對陣列中每一個元素進行處理時，forEach 迴圈更方便。
- 語法：

```
arr.forEach(element => {  
    //對arr陣列的每個元素作動作的程式碼  
});  
arr.forEach(function(element, index){  
    //對arr陣列的每個元素作動作的程式碼  
});
```

- arr: 要處理的陣列

練習

Ex3: 全班有50位同學，假設每位同學各有國文、英文、數學共三科成績，計算每位同學總分與平均，以及各科平均成績。(如下)

編號	國文	英文	數學	總分	平均
1	90	86	75		
2	85	93	86		
...
50	95	74	95		
平均					

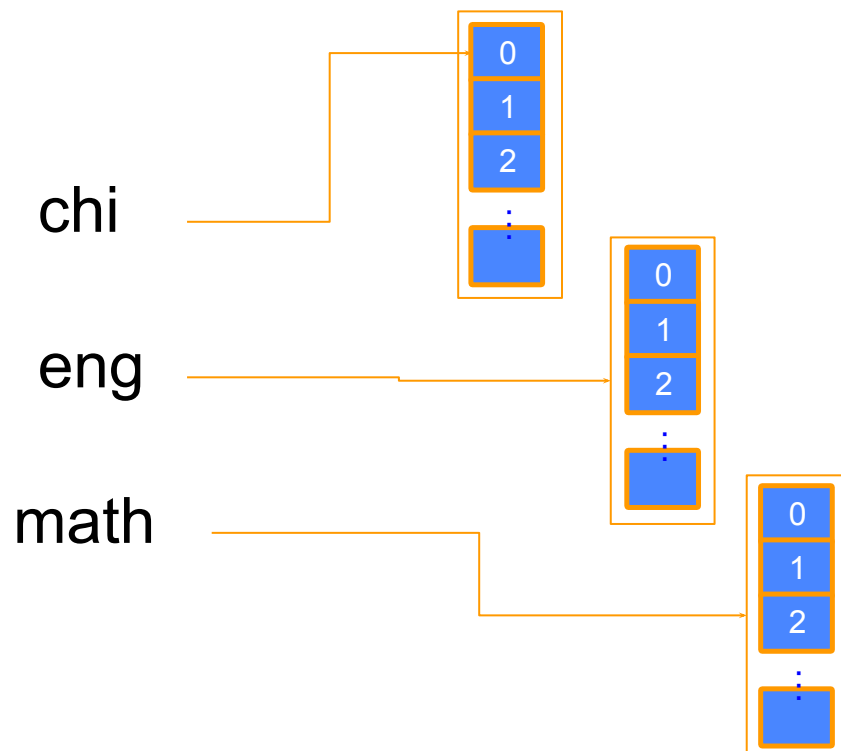
多個一維表單元件陣列

HTML 製作元件陣列

```
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
```

JQuery 接收表單陣列

```
let chi = new Array();
$("input[id=chi]").each(function(index, element){
    chi.push(parseInt($(this).val()));
});
let eng = new Array();
$("input[id=eng]").each(function(index, element){
    eng[i] = parseInt($(this).val());
});
let math = new Array();
$("input[id=math]").each(function(index, element){
    math.push($(element).val());
});
```



分別計算

編號	國文	英文	數學	總分	平均
1	90	86	75		
2	85	93	86		
...
50	95	74	95		
平均					

步驟1: 先分別算出國文, 英文, 數學總和與平均

步驟2: 算出個別學生的總分, 並以`console.log()`顯示結果

步驟3: 將總和結果顯示於HTML元件陣列(參考下頁)

表單元件陣列

HTML 製作元件陣列

<div id="sum"></div>

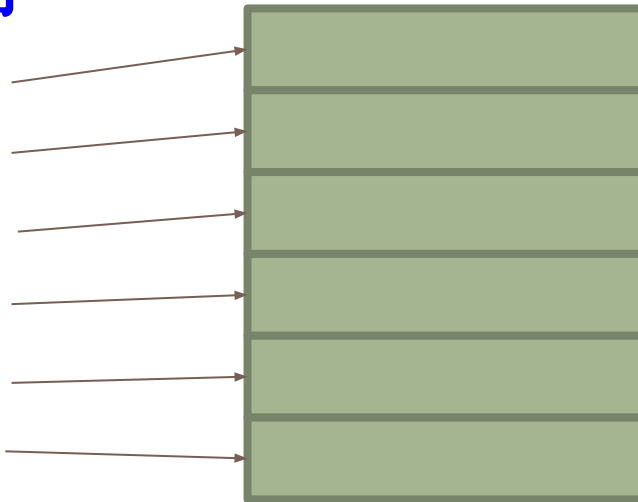
<div id="sum"></div>

<div id="sum"></div>

<div id="sum"></div>

<div id="sum"></div>

<div id="sum"></div>



sum[0]

sum[1]

sum[2]

sum[3]

sum[4]

sum[5]

Javascript 陣列=>表單陣列

```
let sum = new Array();
```

```
$("#div[id=sum]").each(function(index, element){
```

```
    $(this).html(sum[index]);
```

```
});
```

\$(element).html(sum[index]);

多維表單元件陣列 (另一種作法)

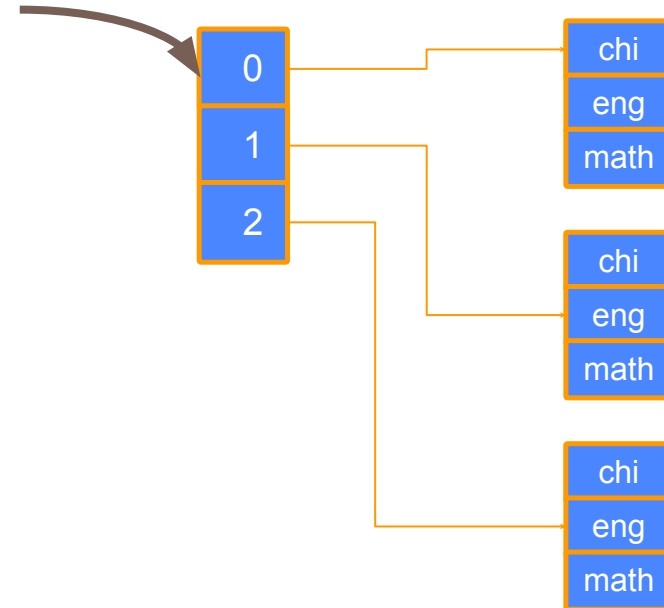
HTML 製作元件陣列

```
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
<input type="text" id="chi" value="1"><input type="text" id="eng" value="2"><input type="text" id="math" value="3"><br>
```

JQuery 接收表單陣列

```
let score = new Array();
$("input[id=chi]").each(function(index, element){
    score[index] = new Array();
    score[index]["chi"] = parseInt($(element).val());
});
$("input[id=eng]").each(function(index, element){
    score[index]["eng"] = parseInt($(element).val());
});
$("input[id=math]").each(function(){
    score[index]["math"] = parseInt($(element).val());
});
```

score



多維表單元件陣列 (另一種作法)

加總

```
let sum = new Array();
let chisum = 0;
let engsum = 0;
let mathsum = 0;
score.forEach(function(element, index){
    chisum += element["chi"];
    engsum += element["eng"];
    mathsum += element["math"];
    sum[index] = element["chi"] + element["eng"] + element["math"];
})
```

最後將chisum, engsum, mathsum, sum[]的結果分別顯示

練習 — 飲料訂購單

品項	價格	數量
奶茶	30	2
炸雞	35	2
薯條	40	2
計算	<input type="radio"/> 會員 <input type="radio"/> 非會員	

↓

親愛的顧客, 你點的是
奶茶: $30 \times 1 = 30$
炸雞: $35 \times 3 = 105$
薯條: $40 \times 2 = 80$
會員八折: $215 \times 0.8 = 172$
總共6份, 共計172元

函式 (Function)

❖ 將一段經常使用的程式碼定義出一個通用的用法及其功能

➤ 例如: 陣列加總、平均

❖ 優點

➤ 提高程式可讀性

➤ 提升寫程式的效率

➤ 讓程式碼可重複使用, 將程式模組化

➤ 提高除錯效率

➤ 各別函數可各別做測試與修正

函式基本語法

```
function 函式名稱(arg1, arg2, ...){  
    程式敘述  
  
    ...  
    return 回傳值;  
}
```

- ❖ 函式名稱的命名規則與變數相同，不可與其他函數或變數同名。
- ❖ `arg1`、`arg2`... 代表要傳入函式的參數。函式的參數可有可無，視程式設計上的需要而定。
- ❖ `return` 會將回傳值傳回給呼叫者。回傳值可以是任何型別，例如字串、數字、變數、陣列、物件等。
- ❖ 如果函式不需要回傳值，則可以省略 `return` 敘述。

函式的呼叫範例

呼叫的程式

```
...  
bmi = getBMI(70, 1.7);  
...
```

被呼叫的函式
getBMI

```
function getBMI(weight, height){  
    const bmi = weight/(height*height);  
    return bmi;  
}
```



函式的參數可以有預設值

```
...  
bmi = getBMI();  
bmi = getBMI(80);  
bmi = getBMI(undefined, 1.8);  
...
```

```
function getBMI(weight=75, height=1.7){  
    const bmi = weight/(height*height);  
    return bmi;  
}
```