



https://jquery.com/



jQuery

jQuery

李建輝 Gary

garylee@ispan.com.tw

1.jQuery OverView

- 什麼是jQuery?
 - jQuery的由來及介紹
- 為什麼使用jQuery?
 - jQuery的特性
- 使用jQuery
 - jQuery預先步驟

什麼是jQuery?

- 由John Resig在2006年1月的BarCamp NYC上釋出第一個版本。
- jQuery 是一個JavaScript函式庫，可讓你在頁面中你可運用它所提供的選擇器(Selector)選取元件(element);再針對這些被選取的元件，利用jQuery提供的功能(API)做一些操作或變更
- jQuery提供許多功能(API)操作你所選取的元件
- 萬能的錢字符 \$

為什麼使用jQuery?

- 簡潔輕量級的 JavaScript Library
- 簡化的 JavaScript 例行工作
- 方便存取或修改部份頁面內容
- 加入特效和動畫效果
- 支援既有 CSS 語法
- 支援 AJAX 功能
- 解決 JavaScript 無法相容於多種 Browser 的問題
- 提供 Extension 機制，方便撰寫 Plugin 新增功能

瀏覽器的支援

• 桌上型瀏覽器

- Chrome: 目前版本及前一個版本
- Firefox: 目前版本及前一個版本, ESR
- Edge: 目前版本及前一個版本
- Safari: 目前版本及前一個版本
- Opera: 目前版本
- Internet Explorer: 9+

• 行動裝置瀏覽器

- Android 4.0+: stock browser
- iOS 7+: Safari

🔍 <https://jquery.com/>



使用jQuery

- 下載 jQuery
- CDN 的引用
- 文件備妥處理器
- jQuery 使用初體驗

下載jQuery

- 標準版(v1,v2,v3)
 - 正常版(Uncompressed)：作為開發或除錯用(可讀性高)
 - 壓縮版(Minified)：由正常版去掉註解、換行符號、空白、Tab等，作為正式上線用(可讀性低)
 - 映射(map)版：壓縮版的javascript code對應到正常版的原始碼
- 精簡版(v3,不支援ajax及effects)
 - 正常版(Uncompressed)：jquery-xxx.slim.js
 - 壓縮版(Minified)：jquery-xxx.slim.min.js
 - 映射(map)版： jquery-xxx.slim.min.map
- 使用方式：
 - `<script src="jquery檔案路徑"> </script>`

CDN引用

- CDN(Content Delivery Network)內容傳遞網路
- jQuery CDN的引用方式
 - `<script src="CDN網址"> </script>`
 - `<script src="http://code.jquery.com/*.min.js"> </script>`
- 其他 CDN :
 - Google CDN
 - Microsoft CDN
 - CDNJS CDN
 - jsDelivr CDN

文件備妥處理器

- 建立HTML DOM Tree後，即可進行操作不需要等待圖片下載完成
- `ready()`

```
<script>  
  jQuery(document).ready(function() {  
    //網頁載入後立刻執行JavaScript程式  
  });  
</script>
```

🔍 <https://jquery.com/>



文件備妥處理器

- 建立HTML DOM Tree後，即可進行操作不需要等待圖片下載完成

```
<script>
  $(function() {
    //網頁載入後立刻執行JavaScript程式
  });
</script>
```

Selector

- jQuery 主要用在DOM物件操作
 - 選取element jQuery的Selector
 - 對所選取的element做動作
- 用jQuery關鍵字使用jQuery Library
 - `jQuery('element').action()`
- `$`代替jQuery
 - `$('element').action()`

鏈式語法 Chaining

- 多個方法套用在相同選擇器上，稱為鍊式語法讓程式碼更簡潔
- **注意**當其中一個方法無法正確執行時，後續方法也不會被執行

chaining 鏈接

- css 語法：
selector.css("proName" [, "value"]) – 讀取或設定樣式
- chaining 語法:
\$(selector).css("proName", "value")
 - .css("proName", "value")
 - .css("proName", "value")



<https://jquery.com/>



使用jQuery



this 及 \$(this)

- this :
 - javascript 物件，代表目前的 DOM 物件
- \$(this) :
 - 選取網頁內的元素，包成陣列型態，再封裝成 jQuery 的物件傳回，

2.jQuery 選擇器Selectors

- jQuery 選擇器 (selectors)
- Basic selector (基本選擇器)
- Basic Filter selector (基本過濾器選擇器)
- Form Selector (表單選擇器)

jQuery selectors

- jQuery的選擇器
是基於現有的CSS選擇器，此外，再新增自訂的選擇器
- 用來選取網頁上的元素，以陣列型態顯示，再封裝成 jQuery 的物件傳回
- jQuery提供許多選擇器api方法
ex:基礎選擇器、屬性選擇器、基本過濾選擇器、表單選擇器...等等

Basic selector(基本選擇器)

- All selector (" * ") : 選取所有元素
- Element Selector("element") : 選取element元素
\$("div") : 選取所有div的元素
- Class selector(".class") : 選取.class屬性的所有元素
\$(".myClass") : 選取所有類別為myClass的元素
- ID Selector("#id") : 選取id屬性的單一元素
\$("#myDiv") : 選取id為myDiv的元素
- Multiple Selector ("selector1,selector2....") : 選擇所有指定選擇器的合併結果
\$("div,span,p.myClass") : 選取所有div, span以及類別為myClass的p元素

Basic Filter selector(基本過濾選擇器)

- **:eq(index)** :

選取符合條件的元素中等於索引值第n個(索引值由0開始)，如`$("td:eq(2)")`

- **:even** :

選取符合條件的元素中偶數的元素，如`$("tr:even()")`

- **:odd** :

選取符合條件的元素中奇數的元素，如`$("tr:odd()")`

- **:lt(index)** :

選取符合條件的元素中小於索引值第n個，如`$("td:lt(3)")`

Basic Filter selector(基本過濾選擇器)

- **:gt(index)** :

選取符合條件的元素中大於索引值第n個，如`$("td:lt(3)")`

- **:first** :

選取符合條件的元素中第一個元素，如`$("tr:first")`

- **:last** :

選取符合條件的元素中最後一個元素，如`$("tr:last")`

- **:not(selector)** :

選取沒有符合條件的元素，如`$("input:not(:checked)")`

Form selector(表單選擇器)

- **:button**

選取所有type是button的元素及button元素，例如：`$(":button")`

- **:input**

選取表單中的所有元素(textarea,select,button)，例如：`$(":input")`

- **:checked**

選取屬性為checked的所有元素，例如：`$(":checked")`

- **:selected**

選取屬性為selected的所有元素，例如：`$(":selected")`

- **:disabled**

選取屬性為disabled的所有元素，例如：`$(":disabled")`

Form selector(表單選擇器)

- **:radio**

選取type是radio的所有元素，例如：`$(":radio")`

- **:checkbox**

選取type是checkbox的所有元素，例如：`$(":checkbox")`

- **:image**

選取type是image的所有元素，例如：`$(":image")`

- **:text**

選取type是text的所有元素，例如：`$(":text")`

Content filter selector(內容過濾選擇器)

- **:contains(text)**

選取所有內文含有指定字串的元素，如 `$("p:contains('Hello'))"`

- **:empty**

選取空的p元素，如 `$("p:empty")`

- **:parent**

選取含有子元素或內文的元素，如 `$("p:parent")`

- **:has(selector)**

選取含有 selector 的元素，如 `$("div:has(p)")`

- **is(selector)**

方法用來判斷目前的元素是否符合某個元素，如果是就會回傳 true。

Child filter selector(子過濾選擇器)

- **:first-child**

選取每個父元素下的第一個子元素，如 `$("div p:first-child")`

- **:last-child**

選取每個父元素下的最後一個子元素，如 `$("div p:last-child")`

- **:nth-child(n)**

選取每個父元素下的第n個子元素，如 `$("div p:nth-child(2)")`

- **:only-child**

選取每個父元素下只有一個子元素，如 `$("div p:only-child")`

Filtering selector

- **filter(selector)**方法用來從找到的元素中，再找出符合 selector
- **first()** 方法用來從找到的元素中，取得第一個元素。
- **eq(index)**方法用來從找到的元素中，取得 index 代表的那一個元素。
- **last()** 方法用來從找到的元素中，取得最後一個元素。
- **index()** 方法用來從找到的元素索引值。

Hierarchy selector(階層選擇器)

- parent > child (子選擇器)
 - 選取parent元素下的直接child元素(往下一層)，如 `$("#div1 > a")`
- ancestor descendant(後代選擇器)
 - 選取ancestor元素下的所有descendant元素(往下所有的層級)，如 `$("#div1 a")`
- prev + next (同層相鄰選擇器)
 - 選取prev元素同一層級相鄰的next元素，如 `$("#div1 + a")`
- prev ~ siblings (同層全體選擇器)
 - 選取prev同一層級的所有siblings元素，如 `$("#div1 ~ a")`



<https://jquery.com/>



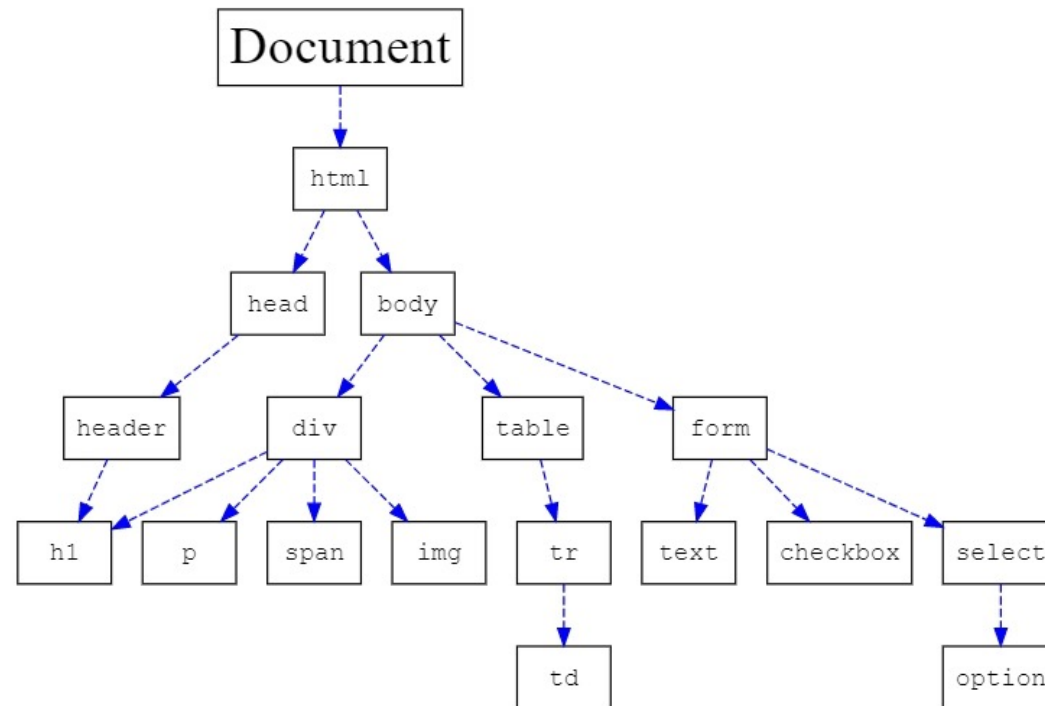
3.jQuery Tree Traversal

- Tree Traversal 與 DOM
- 尋訪子元素方法
- 尋訪父元素方法
- 尋訪同階層方法

Tree Traversal

- DOM : Document Object Model(文件物件模型)
是一組用來處理HTML及XML文件的API
文件中的所有資料，皆可透過DOM來存取、修改、刪除及新增
- Tree Traversal :
透過 DOM Tree 的結構，找到要控制的元素。

Tree Traversal



jQuery Tree Traversal

- **.children()** 方法用來取得目前元素的下一層元素。
- **.find()** 方法用來從目前元素往下找好幾層的元素。

jQuery Tree Traversal

- **.parent()** 方法用來取得目前元素的上一層父元素。
- **.parents()** 方法用來取得目前元素往上的所有父元素直到文件的根元素，在 html 文件中就是 html 元素。
- **.parentsUntil()** 方法用來取得目前元素往上的所有父元素直到碰到符合 selector 條件的父元素停止，取得的元素不包含符合 selector 條件的父元素。
- **.closest()** 方法用來取得目前元素往上找最接近的元素。

jQuery Tree Traversal

- **next()** 方法用來取得目前元素的後一個元素。
- **nextAll()** 方法用來取得目前元素的所有後面元素。
- **nextUntil()** 方法用來取得目前元素的所有後面元素，但不包含符合 `selector` 條件的那一個元素。

jQuery Tree Traversal

- **next()** 方法用來取得目前元素的後一個元素。
- **nextAll()** 方法用來取得目前元素的所有後面元素。
- **nextUntil()** 方法用來取得目前元素的所有後面元素，但不包含符合 `selector` 條件的那一個元素。

jQuery Tree Traversal

- **prev()** 方法用來取得目前元素的前一個元素。
- **prevAll()** 方法用來取得目前元素的所有前面元素
- **prevUntil()** 方法用來取得目前元素的所有前面

jQuery Tree Traversal

- **.end()** 方法用來結束目前的 filtering，回復到上一個的結果。
- **siblings([selector])** 方法用來取得目前元素的前後元素。

4.jQuery Event(事件處理)

- 事件

- 指使用者對瀏覽器或網頁內容所做的某個動作。例如，滑鼠點擊(click)、鍵盤輸入(keydown)等。
- 事件發生時所做的功能，稱為事件處理程序(event handler)，通常是一個函數。

Event Basics 事件基本語法

- 事件基本語法
 - `selector.eventTypeName(function(){....});`
 - `eventTypeName`
 - 滑鼠事件
 - 鍵盤事件
 - 表單事件
 - 視窗事件
 - 文件事件
- `selector.on("eventyTypeName",function(){...})`

Events Type 事件的類型

滑鼠事件

click
dblclick
contextmenu
hover
mousedown
mouseup
mouseenter
mouseleave
mouseover
mouseout
mousemove

鍵盤事件

keydown
keypress
keyup

表單事件

blur
change
focus
focusin
focusout
select
submit

視窗事件

error
resize
scroll

文件事件

load
ready
unload

Event物件

- 何謂 event 物件
- event 物件的屬性
- event 物件的方法
- event 物件範例

何謂Event物件

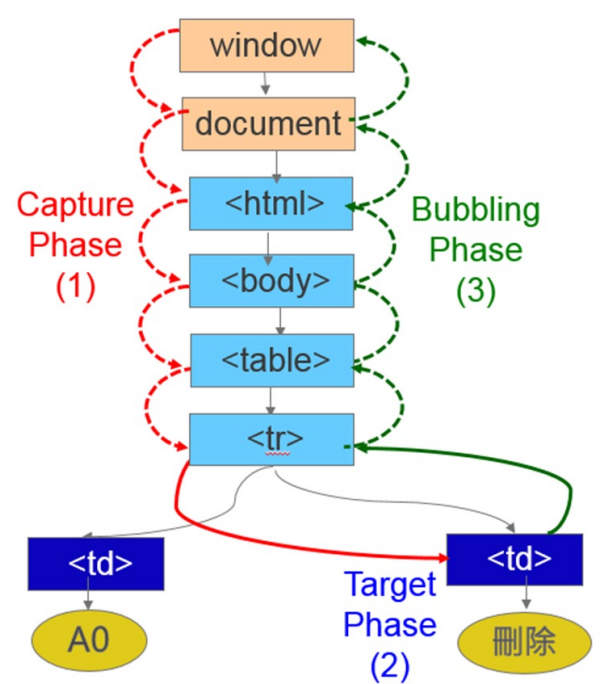
- Event 物件
 - 網頁上的事件觸發的同時會產生event物件。
 - 是由屬性和方法組成

event 物件的屬性

- type : 事件的類型
- which : 回傳鍵盤輸入的編碼或滑鼠按鈕的代碼
- metakey : 回傳是否按下 meta 鍵
- pageX, pageY : 滑鼠游標相對於文件左邊界或上邊界的位置
- target : 回傳觸發事件的 DOM 元素
- currentTarget : 回傳事件正在處理時所在的 DOM 元素
- data : 傳送data給繫結的事件處理程序

event 物件的方法

- `preventDefault()`
 - 停止(取消)預設的動作
- `stopPropagation()`
 - 取消事件氣泡



5.Mouse Events 滑鼠事件

- **click()** : 點擊
- **dblclick()** : 雙點
- **contextmenu()** : 按滑鼠右鍵
- **mousedown()** : 滑鼠在選取的元素上按下
- **mouseup()** : 滑鼠從選取的元素上放開

Mouse Events 滑鼠事件

- **mouseenter()**：只有當滑鼠游標移入選取元素時，才會觸發此事件
- **mouseleave()**：只有當滑鼠游標離開選取元素時，才會觸發此事件
- **mouseover()**：當滑鼠游標移入選取元素及其子元素時，都會觸發此事件
- **mouseout()**：當滑鼠游標離開選取元素及其子元素時，都會觸發此事件
- **mousemove()**：每當滑鼠游標在選取元素及其子元素上移動時，都會觸發此事件
- **hover**：滑鼠游標移入或移出選取元素時，都會觸發此事件

6.Keyboard events (鍵盤事件)

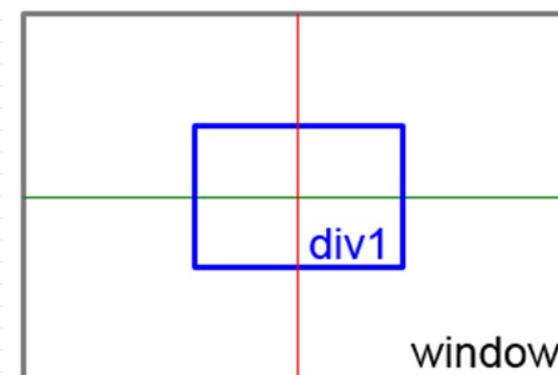
- **keydown()** : 鍵盤按下時觸發
- **keyup()** : 鍵盤離開時觸發
- **keypress()** : 鍵盤按下放開時觸發

7. Form events (表單事件)

- **focus()** : 滑鼠游標移入
- **blur()** : 滑鼠游標離開
- **change()** :
 - 偵測 input 內容修改(修改完游標離開才會觸發事件)及下拉式選單選項

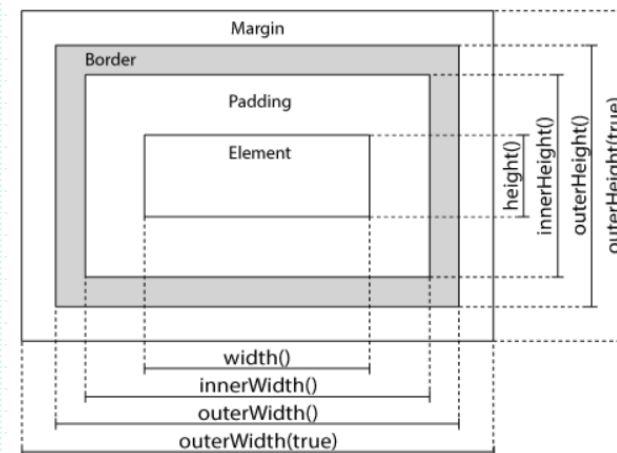
8.Windows events (視窗事件)

- **width**([value]) : 讀取或設定元素的寬度
- **height**([value]) : 讀取或設定元素的高度
- **resize**([handler])事件：偵測瀏覽器調整大小變化



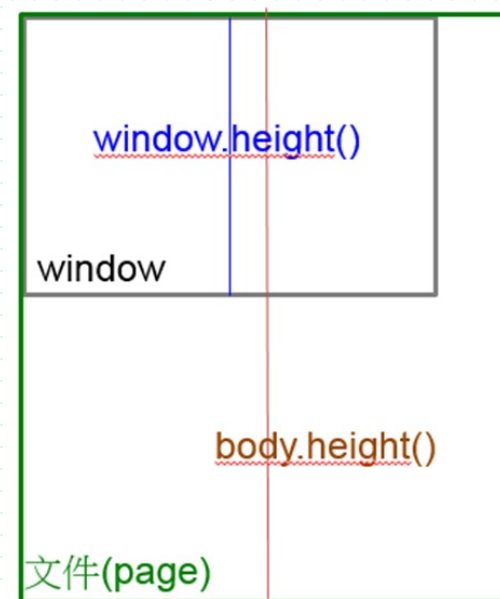
Windows events (視窗事件)

- **innerWidth()** : 讀取元素包含padding的寬度
- **innerHeight()** : 讀取元素包含padding的高度
- **outerWidth()** : 讀取元素含padding和border寬度
- **outerHeight()** : 讀取元素含padding和border高度
- **outerWidth()** : 是否包含元素margin的寬度
- **outerHeight()** : 是否包含元素margin的高度



Windows events (視窗事件)

- **scroll**([handler])：偵測滑鼠滾動
- **scrollTop**([value])：讀取或設定捲軸垂直位置
- **scrollLeft**([value])：讀取或設定捲軸水平位置



9. On binding Event(事件綁定)

- 單一事件連接單一個事件處理函數
- 多個事件連接同一個事件處理函數
- 多個事件連接多個事件處理函數

On binding Event(事件綁定)

- `selector.on(events[,data],handler)`
 - `selector` : 選取的元素。
 - `events` : 觸發事件(如"click", "mouseenter", "resize"等)
 - `data` : 事件發生時，要傳送給事件處理函數(handler)的資料
 - `handle` : 事件處理函數(function({}))

On binding Event(事件綁定)

- **one(events[,data],handler)** : 只會觸發一次的事件
 - `$("#but").one("click",function(){});`
- **off(events[,selector],handler)** :
 - 移除(取消)事件

10.動態綁定

- **selector1.on(events[,selector2][,data],handler)**
 - selector1 : 一開始就在HTML上寫好的元素，不是程式動態產生的元素
 - events : 觸發事件(如"click", "mouseenter", "resize"...等)
 - selector2 : 觸發事件所選取的元素的子元素，可以由程式動態產生的元素
 - data : 傳送給事件處理程序(handler)的資料
 - handler : 事件處理程序(function(){}))

11.jQuery DOM

- 讀取及設定元素內容
- 在選取元素內部新增元素
- 在選取元素外部新增元素

讀取及設定元素內容

- text()方法
- html()方法
- val()方法

text()方法

- **text()**：讀取選取元素文字內容
 - 例如：`$("p").text()`
- **text(text)**：修改選取元素文字內容
 - 例如：`$("p").text("set Value")`

html()方法

- **html()**：讀取選取元素的html標籤及文字內容
 - 例如：`$("p").html()`
- **html(htmlString)**：設定選取元素的html標籤及文字內容
 - 例如：`$("p").html("<h1>setValue</h1>")`

val()方法

- **val()** : 讀取表單元素的值
- **val(value)** : 設定表單每個元素的值

在選取元素內部新增元素

- `append()`方法
- `appendTo()`方法
- `prepend()`方法
- `prependTo()`方法

append()及appendTo()方法

- **append(content)** : 在選取元素內文尾端插入內容
 - `$("p").append("<u>Hello</u>");`
- **appendTo(target)** : 在選取元素內文尾端插入目標內容
 - `$("<p>Test</p>").appendTo(".inner");`

prepend()及prependTo方法

- **prepend(content)** : 在選取元素內文前端插入內容
 - `$(".contain").prepend("<i>JavaScript</i>");`
- **prependTo(target)** : 在選取元素內文前端插入內容
 - `$("<p>Hello world</p>").appendTo(".inner");`

在選取元素外部新增元素

- before()方法
- insertBefore()方法
- after()方法
- insertAfter()方法

after() 及 insertAfter()方法

- **After**(content) : 在選取元素外部後面插入內容
 - `$("p").after("<u>Hello</u>");`
- **insertAfter**(target) : 在選取元素外部後面插入內容
 - `$("<p>Test</p>").insertAfter(".outter");`

before() 及 insertBefore()方法

- **before(content)** : 在選取元素外部前面插入內容
 - `$(".contain").before("<i>JavaScript</i>");`
- **insertBefore(target)** : 在選取元素外部前面插入內容
 - `$("<p>Hello world</p>").insertBefore(".outter");`

12.jQuery DOM II

- 樣式處理
- 讀取及設定元素屬性
- 刪除元素屬性
- 類別屬性

css() 方法

- **css("propertyName")** : 讀取樣式
 - `$("#p").css("color");`
- **css("propertyName", "value")** : 設定樣式
 - `$("#p").css("color", "green");`
- **css(properties)** : 設定多個樣式
 - Properties : `{"propertyName": "value", "propertyName": "value", ...}`
 - `$("#p").css({"color": "red", "background-color": "blue"})`

attr()及removeAttr()方法

- **attr(attributeName)**：讀取符合元素中第一個元素的屬性的值
 - 例如：`$("img").attr("src")`
- **attr(attributeName,value)**：設定每個符合元素的一個或多個屬性的值
 - 例如：`$("img").attr("src", "myimage.jpg")`
- **removeAttr(attributeName)**：移除元素的屬性
 - `$("p").removeAttr("id");`

prop()方法

- **prop(propertyName)**：讀取元素的固有屬性的值
 - 例如：`$("checkbox").prop("checked")`
- **prop(propertyName,value)**：設定元素的固有屬性的值
 - 例如：`$("checkbox").prop("checked","true")`

addClass()及removeClass()方法

- **addClass(className)** : 新增每個符合的元素的類別
 - `$("p").addClass("myclass");`
- **removeClass([className])** : 移除符合的每一個元素的多個或全部類別
 - `$("p").removeClass("myclass");`

toggleClass()及hasClass() 方法

- **toggleClass(className)** : 將符合的元素的類別切換
 - `$("p").toggleClass("myclass")`
- **hasClass(className)** : 判斷元素是否有類別存在
 - `$("p").has("myclass")`

13.其他 jQuery API

- each()方法
- isArray()方法
- map()方法
- grep()方法

each() 方法

- **jQuery.each(function)** 方法是用來將包裝在 jQuery 物件中的元素，一筆一筆的讀出來處理。
 - **function(index, element)** : index 表示 jQuery 物件中元素的位置，從 0 開始計算，element 表示 jQuery 物件中的元素
- **\$.each(array, callback)**
 - **callback** : **function(indexInArray, value)**
- **\$.each(object, callback)**
 - **callback** : **function(propertyName, valueOfProperty**

inArray()方法

- **jQuery.inArray(value, array[, fromIndex])** 方法用來搜尋某值有沒有在陣列中，如果有會傳值所在位置的索引值，如果沒有回傳 -1，索引值時從 0 開始。
 - **value**：用來搜尋陣列中有沒有此值。
 - **array**：被搜尋的陣列。
 - **fromIndex**：陣列的索引值，用來定義從哪個索引值得位置開始搜尋，預設是0，表示要搜尋整個陣列。

grep()方法

- **jQuery.grep(array, function[, invert])** 用來根據 function 中所設定的條件，找到陣列中符合的資料回傳。
 - **array**：用來被搜尋的陣列。
 - **function(elementOfArray, indexInArray)**：此 function 會逐一的讀取陣列中的元素，根據某條件來判斷符合的回傳 **true**，不符合的回傳 **false**。
 - **Invert**：預設是 **false**，表示會回傳符合function 條件的元素。

map()方法

- **jQuery.map(array, callback)** 方法用來逐一的將陣列中的元素讀出來處理，完成後再回傳產生新的陣列。
 - **array**：用來被逐一讀取資料的陣列。
 - **callback(elementOfArray, indexInArray)**：callback 就是一個 function，會逐一的讀取陣列中的元素來處理，並將處理完的元素回傳到新的陣列中。

data()方法

- **data(key,value)** 方法主要是用來透過屬性的方式，儲存一些自訂的資料在元素中。
 - key：自訂的屬性名稱。
 - value：自訂的屬性名稱所對應的值。
- **data(key)** 透過傳入 key 的名稱，讀取其對應的值。

noConflict()方法

- **jQuery.noConflict()**方法是用來處理 \$ 這個別名與其它JavaScript函式庫的衝突問題

14.jQuery Animate

- 基本顯示隱藏
- 滑動效果
- 淡入淡出效果

CSS transition –轉場效果

- transition : 轉場效果
 - 從 A 狀態，轉變成 B 狀態，中間的過程，稱為轉場，CSS稱為 transition
- transition: property duration timing-function delay;
- transition-property : css 的樣式
- transition-duration : 樣式改變時的執行間，單位為秒。
- transition-timing-function : 樣式改變過程中的執行速度及效果
- transition-delay : 多久之後才開始改變樣式

CSS animation

- CSS 的 Animations 主要是透過關鍵影格(keyframes)的設定，可以在動畫執行的過程中，設定不同的動畫規則，語法如下：
animation: name duration timing-function delay iteration-count direction;;
 - duration、timing-function 及 delay 屬性的設定皆同 Transitions的介紹。
 - animation-name：@keyframes的名稱
 - animation-iteration-count：動畫執行的次數，無限次執行用 infinite。
 - animation-direction：動畫執行完成一次後的執行方向，反向用 alternate。

JavaScript動畫寫法

- 使用 `setTimeout()` 或 `setInterval()` 方法。
- 透過改變 CSS 屬性的值，來改變元素的樣式

基本顯示隱藏

- `hide()` 方法
- `show()` 方法
- `toggle()` 方法

hide()方法

- **hide()**方是用來隱藏元素
- **.hide([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **.hide(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **.hide([duration][, easing][, complete])**

show()方法

- **show()**方是用來顯示元素
- **.show([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **.show(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **.show([duration][, easing][, complete])**

toggle()方法

- **toggle()** 方法是用來顯示或隱藏元素
- **.toggle([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **.toggle(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **.toggle([duration][, easing][, complete])**
- **.toggle(display)**

滑動效果

- `slideDown()` 方法
- `slideUp()` 方法
- `slideToggle()` 方法

slideUp()方法

- **slideUp()** 方法是用滑動的方式來隱藏元素
- **slideUp([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **slideUp(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **slideUp([duration][, easing][, complete])**

slideDown()方法

- **slideDown()** 方法是用滑動的方式來顯示元素
- **slideDown([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **slideDown(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **slideDown([duration][, easing][, complete])**

slideToggle()方法

- **slideToggle()** 方法是用滑動的方式來顯示或隱藏元素
- **.slideToggle([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **.slideToggle(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **.slideToggle([duration][, easing][, complete])**

淡入淡出效果

- `fadeIn()` 方法
- `fadeOut()` 方法
- `fadeToggle()` 方法

fadeOut()方法

- **fadeOut()** 方法是用淡化到透明的方式來隱藏元素
- **fadeOut([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **fadeOut(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **fadeOut([duration][, easing][, complete])**

fadeIn()方法

- **fadeIn()** 方法是用淡化到不透明的方式來顯示元素
- **fadeIn([duration][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **fadeIn(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always
- **fadeIn([duration][, easing][, complete])**

fadeToggle()方法

- **fadeToggle()** 方法是用淡化的方式來顯示或隱藏元素
- **.fadeToggle([duration][, easing][, complete])**
 - duration(default:400)：動畫執行的時間
 - complete：動畫完成後會執行的函數(function)
- **.fadeToggle(options)**
 - options：duration、easing、queue、specialEasing、step、progress、complete、start、done、fail、always

15. 自訂動畫 `animate()`

- **`animate()`** 方法是透過樣式的改變，來達到自訂樣式的效果
- `.animate(properties [, duration] [, easing] [, complete])`
 - `properties` : CSS 的屬性及值，表示方式為 `{width:"300px"}`
 - `duration(default:400)` : 動畫執行的時間
 - `easing` : 動畫執行過程中的執行速度及效果
 - `complete` : 動畫完成後會執行的函數(function)

🔍 <https://jquery.com/>



animate.css

- **Animate.css**

<https://animate.style/>

- **Animista**

<https://animista.net/>

16.jQuery UI

- 什麼是 jQuery UI
- jQuery UI 效果
- 開始使用 jQuery UI

🔍 <https://jquery.com/>



什麼是 jQuery UI

- jQuery UI 是以 jQuery 為基礎所建置出的函式庫，提供了具備使用者介面、樣式及程式完整的功能，讓開發工程師可以快速的建置網頁應用。

17. Another Plugins

- lightbox2
- vegas.js
- sweetalert2
- datatable
- timeline
- Masonry
- wow.js
- lightSlider
- tilt.js
- owl carousel.js
- textillate.js

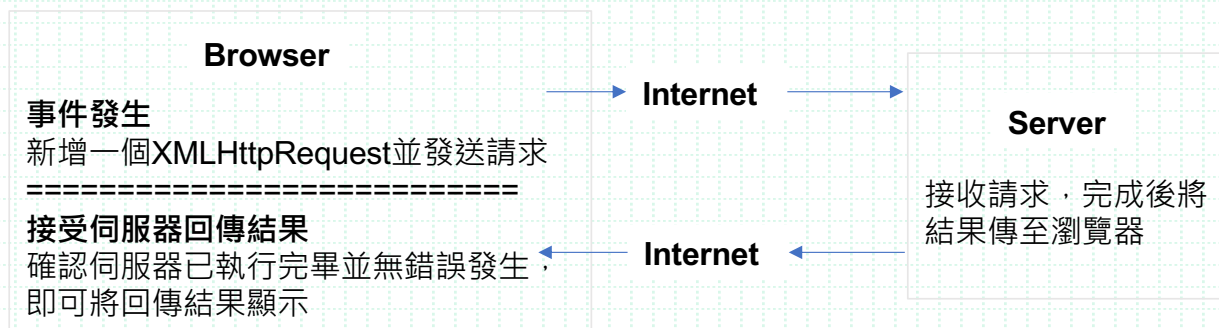


https://jquery.com/



18.jQuery AJAX

- AJAX即「Asynchronous JavaScript and XML」



jQuery AJAX

- **\$.ajax([settings])**
 - url 指定要進行呼叫位址
 - contentType：前端傳送至後端資料型態 application/json text/html
 - data：要傳送至後端的資料
 - dataType：後端返回前端的資料型態 xml html json text
 - type：請求方式 POST GET
 - success：當請求成功時所執行的函數
 - error：當請求失敗時所執行的函數