

jQuery 初探 (Basics)

jQuery 所有的屬性及函式都是定義在「**jQuery**」這個物件之下，這使你不會因為使用 jQuery Library 而與原本有的全域變數等其它命名空間產生衝突。此外，要取得 jQuery 物件也可以透過它提供的另外一個縮寫符號（別名）－錢字號「**\$**」，這時大家可能會問那我有使用其它的 JavaScript Library 也是用「\$」怎麼辦？有辦法，用下面這一行就解決了：

```
jQuery.noConflict();
```

接著你就可以繼續用 \$ 來操作你原本有的函式。但是如果你還是比較想用 \$ 來操作 jQuery 怎麼辦？也有取巧的辦法：

```
(function($) {  
    // 在此區塊內我們使 $ 參照 jQuery 物件  
    // 在此區塊內使用 $ 不會與其它函式衝突  
})(jQuery);
```

你也可以自己取個 jQuery Object 的別名：

```
var $alias = jQuery.noConflict();
```

接下來你就可以使用 \$alias 取代 \$。

jQuery 的基本操作概念 (Coding basics)

jQuery 程式碼由 \$ (或jQuery) 開始 → 後面會接著圓刮號「**()**」→ 而圓刮號裡面的參數是你想叫 jQuery 幫你找什麼 (取得哪個(些)元素) → 接著是你想叫 jQuery 執行什麼動作 (或處理事件)。
。例如：

```
/
$( "#demo" ) : 選取id為demo的元素，並綁定onclick事件
.css( "background-color", "gray" ) :
    叫jQuery將其CSS的背景顏色屬性改成灰色
    你可以用滑鼠點這個範例看效果
/

$( "#demo" ).click( function() {
    $( "#demo" ).css( "background-color", "gray" );
} );
```

Chaining – 串接

在 jQuery 中，幾乎所有成員都會返回自己執行後的結果 – 也是一個 jQuery 物件，因此你可以連續地使用函數 (Chaining)。以下我們用一個範例來說明 Chaining 是怎麼一回事：

```
$( "#container" ).css( "color", "blue" ).css( "background-color", "red" );
```

上面這段程式碼由兩段函式組成：

```
$( "#container" ).css( "color", "blue" ) → 將文字改成藍色
$( "#container" ).css( "background-color", "red" ) → 再將背景顏色改為紅色
```

利用 Chaining 只剩一行，是不是簡潔多了？