網頁設計基礎

[1 HTML 6](#_Toc25784686)

[1.1 HTML簡介 6](#_Toc25784687)

[1.2 HTML文件的基本架構 6](#_Toc25784688)

[1.3 HTML元素的組成 7](#_Toc25784689)

[1.4 DOM文件物件模型簡介 8](#_Toc25784690)

[1.5 HTML常用標籤 9](#_Toc25784691)

[1.5.1 <h1> ~ <h6> 9](#_Toc25784692)

[1.5.2 <br> 9](#_Toc25784693)

[1.5.3 <p> 10](#_Toc25784694)

[1.5.4 <hr> 11](#_Toc25784695)

[1.5.5 <div> 11](#_Toc25784696)

[1.5.6 <span> 12](#_Toc25784697)

[1.5.7 <ul> <li> 12](#_Toc25784698)

[1.5.8 <ol> <li> 13](#_Toc25784699)

[1.5.9 <form>表單相關元素 13](#_Toc25784700)

[1.5.9.1 <form> 13](#_Toc25784701)

[1.5.9.2 <label> 14](#_Toc25784702)

[1.5.9.3 <input> 14](#_Toc25784703)

[1.5.9.4 <fieldset> 16](#_Toc25784704)

[1.5.9.5 <legend> 16](#_Toc25784705)

[1.5.9.6 範例I：html\_form1.html 16](#_Toc25784706)

[1.5.9.7 範例II：html\_form2.html 17](#_Toc25784707)

[1.5.10 <select> <option> 19](#_Toc25784708)

[1.5.11 <textarea> 19](#_Toc25784709)

[1.5.12 <button> 20](#_Toc25784710)

[1.5.13 <img> 21](#_Toc25784711)

[1.5.14 <a> 21](#_Toc25784712)

[1.5.15 <audio> 22](#_Toc25784713)

[1.5.16 <video> 22](#_Toc25784714)

[1.5.16.1 範例：html\_media.html 23](#_Toc25784715)

[1.5.17 <table>表格相關元素 24](#_Toc25784716)

[1.5.17.1 <table> 24](#_Toc25784717)

[1.5.17.2 <caption> 24](#_Toc25784718)

[1.5.17.3 <tr> 24](#_Toc25784719)

[1.5.17.4 <th> 24](#_Toc25784720)

[1.5.17.5 <td> 24](#_Toc25784721)

[1.5.17.6 範例I：html\_table.html 25](#_Toc25784722)

[2 CSS 26](#_Toc25784723)

[2.1 CSS簡介 26](#_Toc25784724)

[2.2 CSS樣式表規則 26](#_Toc25784725)

[2.2.1 行內樣式表(In-line Style Sheets) 26](#_Toc25784726)

[2.2.2 內嵌式樣式表(Embedded Style Sheets) 26](#_Toc25784727)

[2.2.3 外部連結式樣式表(External Style Sheets) 26](#_Toc25784728)

[2.3 CSS選擇器 27](#_Toc25784729)

[2.4 CSS樣式屬性摘要 33](#_Toc25784730)

[2.4.1 字型相關屬性 33](#_Toc25784731)

[2.4.2 文字相關屬性 36](#_Toc25784732)

[2.4.3 盒子模型(Box Model) 36](#_Toc25784733)

[2.4.4 背景相關屬性 38](#_Toc25784734)

[2.4.5 Display(元素顯示)屬性 39](#_Toc25784735)

[2.4.6 定位相關屬性 39](#_Toc25784736)

[2.4.7 Cursor(滑鼠游標)屬性 40](#_Toc25784737)

[2.5 CSS選擇器優先順序 43](#_Toc25784738)

[2.5.1 選擇器具體性(selector's specificity)計分規則 43](#_Toc25784739)

[2.5.2 選擇器具體性計分範例 43](#_Toc25784740)

[3 JavaScript 45](#_Toc25784741)

[3.1 JavaScript基本語法 45](#_Toc25784742)

[3.2 變數 45](#_Toc25784743)

[3.2.1 宣告變數 45](#_Toc25784744)

[3.2.2 變數名稱基本規則 46](#_Toc25784745)

[3.2.3 變數自動宣告 46](#_Toc25784746)

[3.3 JavaScript支援的常用資料型別 46](#_Toc25784747)

[3.3.1 原始資料型別(Primitive Data Type) 46](#_Toc25784748)

[3.3.1.1 字串型別(String) 46](#_Toc25784749)

[3.3.1.2 數值型別(Number) 47](#_Toc25784750)

[3.3.1.3 布林型別(Boolean) 47](#_Toc25784751)

[3.3.1.4 Null型別 47](#_Toc25784752)

[3.3.1.5 未定義型別(Undefined) 47](#_Toc25784753)

[3.3.2 物件(Object) 47](#_Toc25784754)

[3.4 運算子 47](#_Toc25784755)

[3.4.1 算術運算子 47](#_Toc25784756)

[3.4.2 字串串連運算子 48](#_Toc25784757)

[3.4.3 比較運算子 48](#_Toc25784758)

[3.4.4 邏輯運算子 48](#_Toc25784759)

[3.4.5 條件運算子 49](#_Toc25784760)

[3.4.6 指定運算子 49](#_Toc25784761)

[3.5 運算子優先順序 49](#_Toc25784762)

[3.6 流程控制結構 50](#_Toc25784763)

[3.6.1 選擇性結構 50](#_Toc25784764)

[3.6.1.1 if 敘述 50](#_Toc25784765)

[3.6.1.2 if...else敘述 51](#_Toc25784766)

[3.6.1.3 switch敘述 52](#_Toc25784767)

[3.6.2 重複性結構 52](#_Toc25784768)

[3.6.2.1 for loop (for迴圈) 52](#_Toc25784769)

[3.6.2.2 while loop (while迴圈) 53](#_Toc25784770)

[3.6.2.3 do while loop 53](#_Toc25784771)

[3.6.2.4 for...In loop 53](#_Toc25784772)

[3.7 陣列 54](#_Toc25784773)

[3.7.1 建立陣列 54](#_Toc25784774)

[3.7.1.1 使用Array建構方法 54](#_Toc25784775)

[3.7.1.2 使用Array Literal 54](#_Toc25784776)

[3.7.2 存取陣列元素 54](#_Toc25784777)

[3.7.2.1 存取特定位置之陣列元素 54](#_Toc25784778)

[3.7.2.2 使用for loop巡覽每一個陣列元素 54](#_Toc25784779)

[3.7.2.3 使用for in loop巡覽每一個陣列元素 55](#_Toc25784780)

[3.8 函數 55](#_Toc25784781)

[3.8.1 一般函數 55](#_Toc25784782)

[3.8.1.1 函數定義 55](#_Toc25784783)

[3.8.1.2 函數範例 55](#_Toc25784784)

[3.8.2 function literal(無名函數) 56](#_Toc25784785)

[3.8.2.1 建立函數 56](#_Toc25784786)

[3.8.2.2 函數範例 56](#_Toc25784787)

[3.9 物件深入探析 57](#_Toc25784788)

[3.9.1 建立物件 57](#_Toc25784789)

[3.9.1.1 使用JS內建Object建構子函數(Constructor) 57](#_Toc25784790)

[3.9.1.2 使用Object Literal語法 58](#_Toc25784791)

[3.9.1.3 使用使用者自訂建構子函數(Constructor) 58](#_Toc25784792)

[3.9.2 使用物件 58](#_Toc25784793)

[4 jQuery 61](#_Toc25784794)

[4.1 jQuery 簡介 61](#_Toc25784795)

[4.2 jQuery選擇器 61](#_Toc25784796)

[4.2.1 jQuery選擇器列表 61](#_Toc25784797)

[4.2.2 jQuery選擇器使用簡例(jq\_T01.html) 65](#_Toc25784798)

[4.3 jQuery常用API 66](#_Toc25784799)

[4.3.1 存取元素之CSS樣式 66](#_Toc25784800)

[4.3.1.1 .css( propertyName, value ) Returns: jQuery 66](#_Toc25784801)

[4.3.1.2 .addClass( className ) Returns: jQuery 67](#_Toc25784802)

[4.3.1.3 .removeClass( [className ]) Returns: jQuery 67](#_Toc25784803)

[4.3.1.4 範例：jq\_T02.html 67](#_Toc25784804)

[4.3.2 存取元素之屬性 69](#_Toc25784805)

[4.3.2.1 .attr( attributeName ) Returns: String 69](#_Toc25784806)

[4.3.2.2 .attr(attributeName, value) Returns: jQuery 69](#_Toc25784807)

[4.3.2.3 範例：jq\_T03.html(page I) 69](#_Toc25784808)

[4.3.3 巡覽jQuery集合物件內的每一元素 71](#_Toc25784809)

[4.3.3.1 .attr( attributeName ) Returns: String 71](#_Toc25784810)

[4.3.3.2 .each( function ) Returns: jQuery 71](#_Toc25784811)

[4.3.3.3 .html( ) Returns: String 71](#_Toc25784812)

[4.3.3.4 .html( htmlString ) Returns: jQuery 71](#_Toc25784813)

[4.3.3.5 範例：jq\_T03.html(page II) 71](#_Toc25784814)

[4.3.4 事件處理 73](#_Toc25784815)

[4.3.4.1 .ready( handler ) Returns: jQuery 73](#_Toc25784816)

[4.3.4.2 .click( handler ) Returns: jQuery 73](#_Toc25784817)

[4.3.4.3 範例：jq\_T03.html(page III) 73](#_Toc25784818)

[4.3.5 特效與動畫 75](#_Toc25784819)

[4.3.5.1 .show() Returns: jQuery 75](#_Toc25784820)

[4.3.5.2 .hide( ) Returns: jQuery 75](#_Toc25784821)

[4.3.5.3 .fadeIn( [duration ] [, complete ] ) Returns: jQuery 76](#_Toc25784822)

[4.3.5.4 .slideDown( [duration ] [, complete ] ) Returns: jQuery 76](#_Toc25784823)

[4.3.5.5 .fadeOut( [duration ] [, complete ] ) Returns: jQuery 76](#_Toc25784824)

[4.3.5.6 .slideUp( [duration ] [, complete ] ) Returns: jQuery 76](#_Toc25784825)

[4.3.5.7 .animate( ) Returns: jQuery 76](#_Toc25784826)

[4.3.5.8 範例：jq\_T04.html 78](#_Toc25784827)

[4.3.6 使用AJAX與網站伺服器互動 81](#_Toc25784828)

[4.3.6.1 AJAX簡介 81](#_Toc25784829)

[4.3.6.2 jQuery.ajax( [settings ] ) Returns: jqXHR 81](#_Toc25784830)

[4.3.6.3 範例：jq\_T05.html(page I) 82](#_Toc25784831)

[4.3.6.4 以AJAX向伺服器端請求JSON資料 85](#_Toc25784832)

[4.3.6.5 jQuery.getJSON( url [, data ] [, success ] ) Returns: jqXHR 87](#_Toc25784833)

[4.3.6.6 範例：jq\_T05.html(page II) 87](#_Toc25784834)

[4.4 jQuery UI常用元件 89](#_Toc25784835)

[4.4.1 DatePicker(日期選擇器) 89](#_Toc25784836)

[4.4.2 Dialog(對話框) 90](#_Toc25784837)

[4.4.3 Accordion(手風琴選單) 91](#_Toc25784838)

[4.4.4 Draggable 與Droppable 92](#_Toc25784839)

[4.4.5 ToolTip(工具提示框) 93](#_Toc25784840)

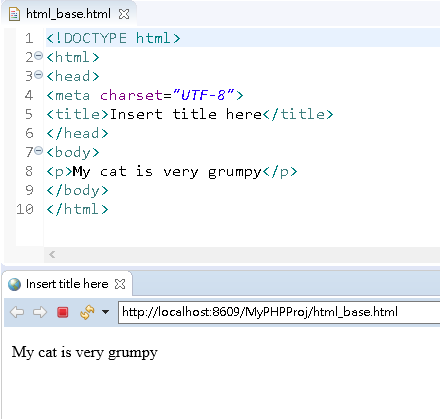
[4.4.6 AnythingSlider(內容輪播器/Content Slider)外掛程式 94](#_Toc25784841)

[4.4.7 範例：jq\_T06.html 95](#_Toc25784842)

# HTML

## HTML簡介

* HTML(HyperText Markup Language/超文字標記語言)是一種用於建立網頁的標記語言。
* HTML是一種基礎技術，常與[CSS](https://zh.wikipedia.org/wiki/CSS)、[JavaScript](https://zh.wikipedia.org/wiki/JavaScript)一起被眾多網站用於設計網頁、網頁應用程式以及行動應用程式的使用者介面。
* HTML以標籤(Tag)將網頁文件不同部分的內容包裹起來成為一系列的元素(elements)，透過瀏覽器的解譯，使得它們得以呈現出不同風格樣式的視覺化網頁。
* HTML文件內容對應透過瀏覽器所呈現出來的視覺化網頁，分別如下之上半部與下半部)：



* 自1996年起，HTML規範一直由眾多商業軟體廠商出資成立的[**全球資訊網協會**](https://zh.wikipedia.org/wiki/%E4%B8%87%E7%BB%B4%E7%BD%91%E8%81%94%E7%9B%9F)**(**World Wide Web Consortium /W3C)所維護。
* 2004年由Apple公司、Mozilla基金會和Opera軟體公司所組成的[**網頁超文字應用技術工作小組**](https://zh.wikipedia.org/wiki/%E7%B6%B2%E9%A0%81%E8%B6%85%E6%96%87%E6%9C%AC%E6%8A%80%E8%A1%93%E5%B7%A5%E4%BD%9C%E5%B0%8F%E7%B5%84)**(**Web Hypertext Application Technology Working Group/WHATWG）開始開發HTML5，並在2008年與W3C共同交付，2014年10月28日完成標準化。

## HTML文件的基本架構

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

</body>

</html>

* <!DOCTYPE html> — 文件類型

在 HTML 發展初期（約莫 1991~2 的年代），文件類型是用來連結一些應遵守的規則，有點像自動校正的功能。然而，現在大家其實不太管文件類型。

* <html></html> — <html> 元素

又被視為根元素（root element），包含了所有顯示在這個頁面上的內容。

* <head></head> —  <head> 元素

裡面放的是你想涵括的重要資訊，但不會顯示於網頁瀏覽者眼前的。

* <body></body> —  <body> 元素

包含了所有會顯示於網頁瀏覽者眼前的內容。 無論是文字、圖片、影片...等。

* <meta charset="utf-8">

這個元素指定了你的文件使用utf-8這種字元編碼， 建議大家都要使用這個元素，它會幫助你免去許多文字無法正確呈現的煩惱。

* <title></title>

呈現於網頁瀏覽者眼前的網頁標題。

## HTML元素的組成





## DOM文件物件模型簡介

* 文件物件模型(Document Object Model / DOM)是一個表達、存取網頁中物件的介面(API/Application Programming Interface)。
* DOM是全球資訊網協會（World Wide Web Consortium，W3C）所制定的一套標準，大部分的瀏覽器開發廠商都會遵循並採用這套標準。
* 有了DOM， JavaScript就可以和網頁上的HTML進行互動，並修改HTML內容。
* DOM將一個HTML網頁的資料結構以樹狀(tree)圖來表達，樹狀圖中的每一塊狀項目稱為節點(nodes)
* HTML DOM樹狀圖範例

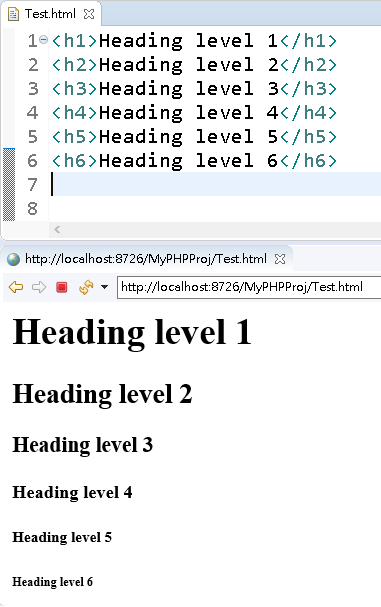


|  |
| --- |
| <html>  <head>  <title>My title</title>  </head>  <body>  <a href="t1.html">My link</a>  <h1>My header</h1>  </body>  </html> |

## HTML常用標籤

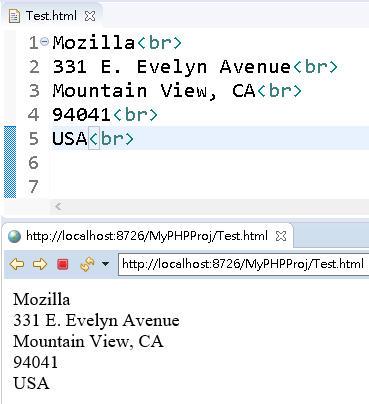
### <h1> ~ <h6>

* The HTML <h1>~<h6> elements represent six levels of section headings (標題).
* <h1> is the highest section level and <h6> is the lowest.



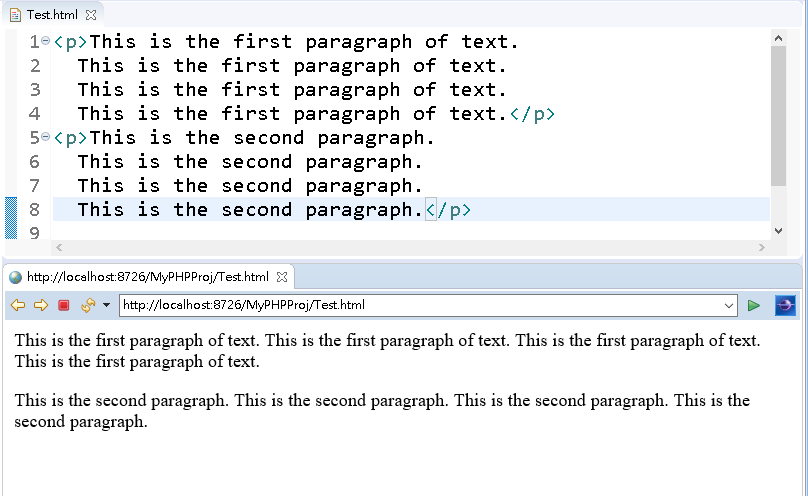
### <br>

* The HTML <br> element produces a line break in text (carriage-return).



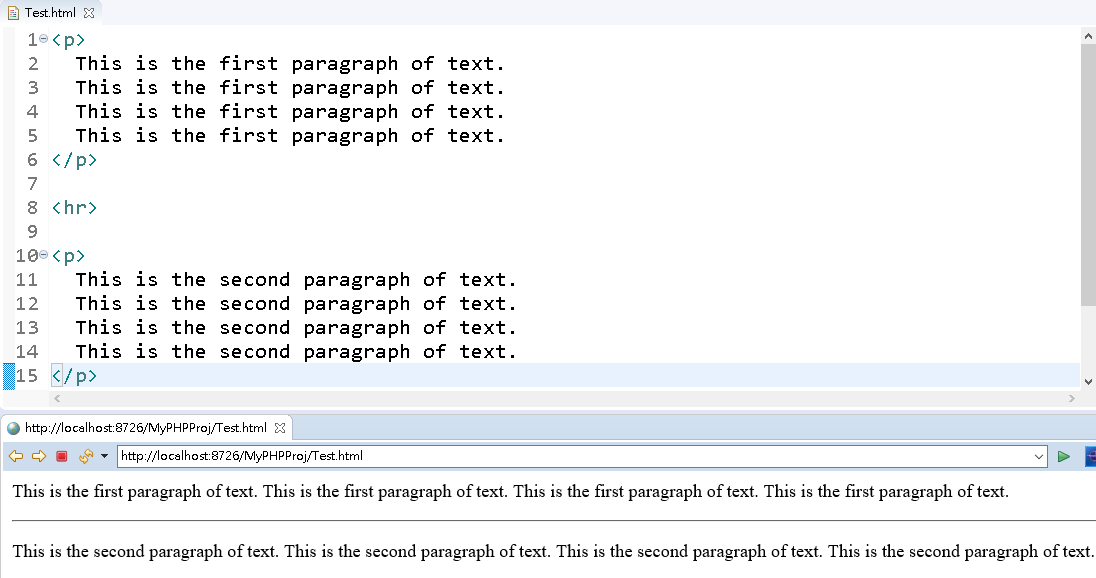
### <p>

* The HTML <p> element represents a paragraph.
* Paragraphs are usually represented in visual media as blocks of text separated from adjacent blocks by blank lines and/or first-line indentation, but HTML paragraphs can be any structural grouping of related content, such as images or form fields.
* Paragraphs are [block-level elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements), and notably will automatically close if another block-level element is parsed before the closing </p> tag.



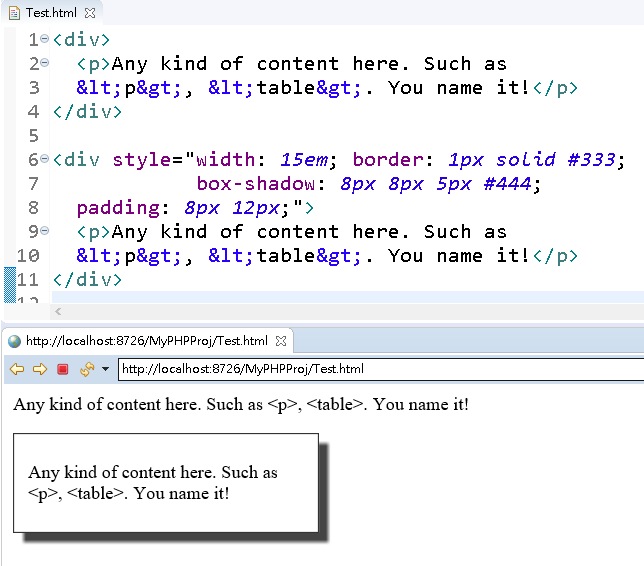
### <hr>

* The HTML <hr> element represents a thematic(主題的) break between paragraph-level elements: for example, a change of scene in a story, or a shift of topic within a section.



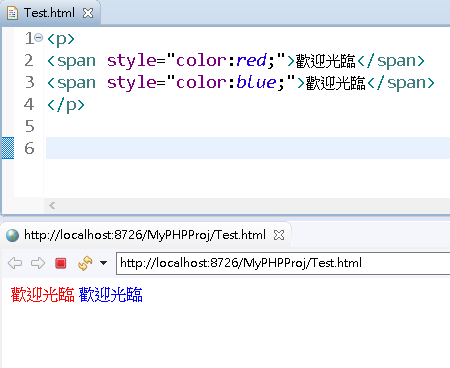
### <div>

* The HTML Content Division element (<div>) is the generic container for flow content.
* It has no effect on the content or layout until styled using [CSS](https://developer.mozilla.org/en-US/docs/Glossary/CSS).



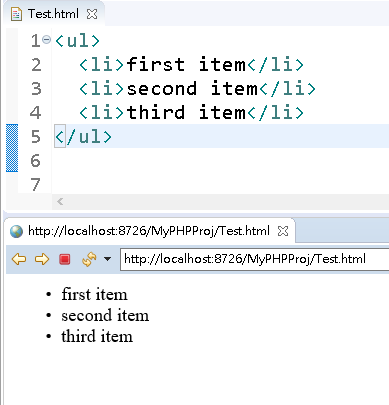
### <span>

* The HTML <span> element is a generic inline container for phrasing(表達) content, which does not inherently(天性地) represent anything.
* It can be used to group elements for styling purposes (using the [class](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes#attr-class) or [id](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes#attr-id) attributes).
* <span> is very much like a [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div) element, but [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div) is a [block-level element](https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements) whereas a <span> is an [inline element](https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements).



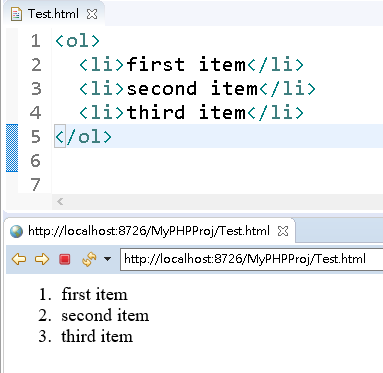
### <ul> <li>

* The HTML <ul> element represents an unordered list of items, typically rendered as a bulleted list.
* The HTML <li> element is used to represent an item in a list.
* It must be contained in a parent element: an ordered list ([<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol)), or an unordered list ([<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul)).



### <ol> <li>

* The HTML <ol> element represents an ordered list of items, typically rendered as a numbered list.



### <form>表單相關元素

#### <form>

* The HTML <form> element represents a document section that contains interactive controls for submitting information to a web server.
  + - action屬性

The URI of a program that processes the form information.

* + - method屬性

The [HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP) method that the browser uses to submit the form. Possible values are:

* post:
* Corresponds to the HTTP [POST method](https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.5) ;
* form data are included in the body of the form and sent to the server.
* get:
* Corresponds to the HTTP [GET method](https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.3);
* form data are appended to the action attribute URI with a '?' as separator, and the resulting URI is sent to the server.
* Use this method when the form has no side-effects and contains only ASCII characters

#### <label>

* The HTML <label> element represents a caption for an item in a user interface.

#### <input>

* + The HTML <input> element is used to create interactive controls for web-based forms in order to accept data from the user.
  + How an <input> works varies considerably depending on the value of its type attribute. The available types are as follows:
    - text:

A single-line text field.

* + - password:

A single-line text field whose value is obscured(隱匿的).

Use the maxlength and minlength attributes to specify the maximum length of the value that can be entered.

* + - checkbox:

A check box(核取方塊) allowing single values to be selected/deselected.

* + - radio:

A radio button(選項按鈕), allowing a single value to be selected out of multiple choices.

* + - submit:

A button that submits the form.

* + - reset:

A button that resets the contents of the form to default values.

* + - button:

A push button with no default behavior.

* + - file:

A control that lets the user select a file.

Use the accept attribute to define the types of files that the control can select.

例如：

<input type="file" accept="audio/\*,video/\*,image/\*,.jpg,.png" />

* + - hidden:

A control that is not displayed but whose value is submitted to the server.

* + - image:

A graphical submit button.

You must use the src attribute to define the source of the image and the alt attribute to define alternative text.

You can use the height and width attributes to define the size of the image in pixels.

* + - color:

A control for specifying a color. A color picker's UI has no required features other than accepting simple colors as text.(如：#ff0000 提交值：%23ff0000)

* + - date:

A control for entering a date (year, month, and day, with no time).(如: 2019-05-20)

* + - time:

A control for entering a time value with no time zone.(如：13:10:20 提交值：13%3A10%3A20 即)

* + - datetime-local:

A control for entering a date and time, with no time zone.(如：2019-10-20T13:10:20 提交值：2019-10-20T13%3A10%3A20)

* + - month:

A control for entering a month and year, with no time zone.(如： 2019-05)

* + - number:

A control for entering a number.

* + - range:

A control ( like a slider(滑桿) control) for entering a number whose exact value is not important.

* + - tel:

A control for entering a telephone number.(註：pattern屬性可設定格式)

* + - url:

A field for entering a URL.

例如：http://www.example.com/index.html

a protocol (http) + a hostname (www.example.com ) + a file name (index.html).

* + - email:

A field for editing an e-mail address.(如：abc#@gmail.com)

#### <fieldset>

* The HTML <fieldset> element is used to group several controls as well as labels ([<label>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label)) within a web form.

#### <legend>

* The HTML <legend> element represents a caption for the content of its parent [<fieldset>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/fieldset).

#### 範例I：html\_form1.html

<form action=*"html\_form1.html"* method=*"get"*

style="width:*400px*;margin:*50px auto*; padding:*10px 30px*;

border:*1px solid #444*;">

<p>

<label for=*"username"*>使用者名稱</label>

<input type=*"text"* id=*"username"* name=*"username"*

value=*"小丸子"* />

</p>

<p>

<label for=*"pwd"*>密碼</label>

<input type=*"password"* id=*"pwd"* name=*"pwd"* />

</p>

<fieldset>

<legend>性別</legend>

<input type=*"radio"* id=*"male"* name=*"sex"*

value=*"Male"* checked=*"checked"* />

<label for=*"male"*>男性</label>

<input type=*"radio"* id=*"female"* name=*"sex"*

value=*"Female"*/>

<label for=*"female"*>女性</label>

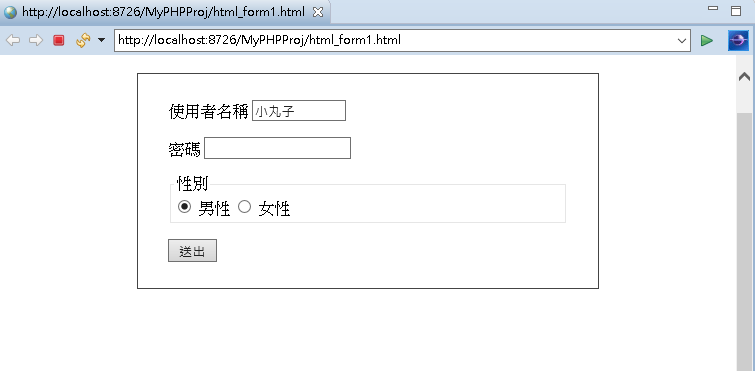
</fieldset>

<p>

<input type=*"submit"* value=*"送出"*/>

</p>

</form>



#### 範例II：html\_form2.html

<form action=*"html\_form2.html"* method=*"get"*>

<fieldset>

<legend>興趣</legend>

<input type=*"checkbox"* id=*"poem"*

name=*"hobby"* value=*"Poem"* />

<label for=*"poem"*>吟詩</label>

<input type=*"checkbox"* id=*"piano"* name=*"hobby"*

value=*"Piano"* /> <label for=*"piano"*>彈琴</label>

</fieldset>

<p>

選色(color): <input type=*"color"* name=*"favcolor"*

value=*"#ff0000"* />

</p>

<p>

生日(date): <input type=*"date"* name=*"bdate"*

value=*"2019-05-20"* />

</p>

<p>

會議時間(time): <input type=*"time"* name=*"mtime"*

value=*"13:10:20"* />

</p>

<p>

會議日期時間(datetime-local): <input type=*"datetime-local"*

name=*"mdt"* value=*"2019-10-20T13:10:20"* />

</p>

<p>

生日年月(month): <input type=*"month"* name=*"bdate2"*

value=*"2019-05"* />

</p>

<p>

數量 (number): <input type=*"number"* name=*"qty"*

min=*"1"* max=*"5"* value=*"1"* />

</p>

<p>

點數(range): <input type=*"range"* name=*"points"*

min=*"0"* max=*"10"* step=*"1"* />

</p>

<p>

TEL(格式:0[1-9]-\d{7,8}): <input type=*"tel"* name=*"phone"*

pattern=*"0[1-9]-\d{7,8}"* value=*"07-1234567"* />

</p>

<p>

URL: <input type=*"url"* name=*"url"*

value=*"http://www.iii.org.tw"* />

</p>

<p>

Email: <input type=*"email"* name=*"email"*

value=*"abc#@gmail.com"* />

</p>

<p>

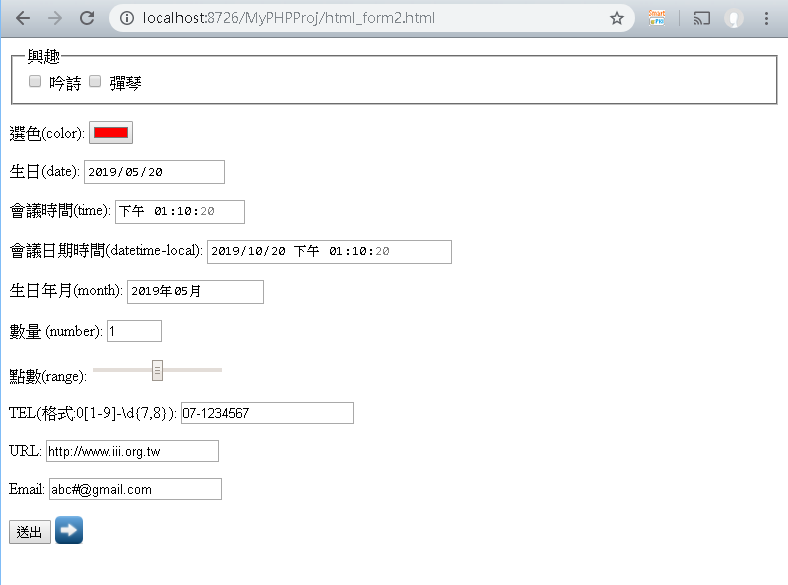
<input type=*"submit"* value=*"送出"* />

<input type=*"image"* src=*"images/submit64.png"*

width=*"28"* height=*"28"* />

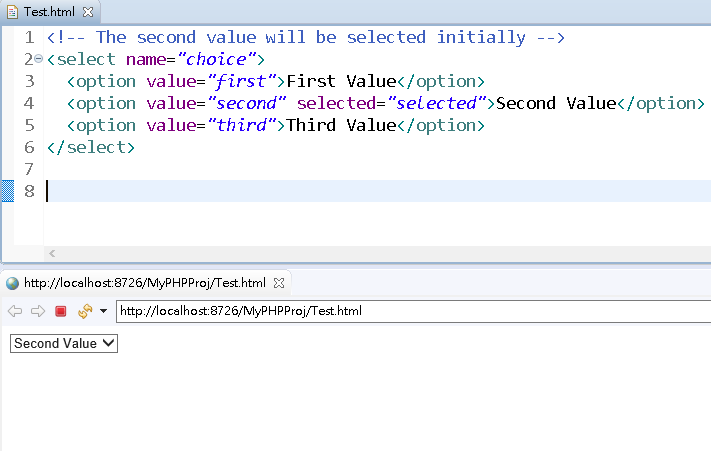
</p>

</form>



### <select> <option>

* The HTML <select> element represents a control that provides a menu of options.
* The HTML <option> element is used to define an item contained in a <select>.



### <textarea>

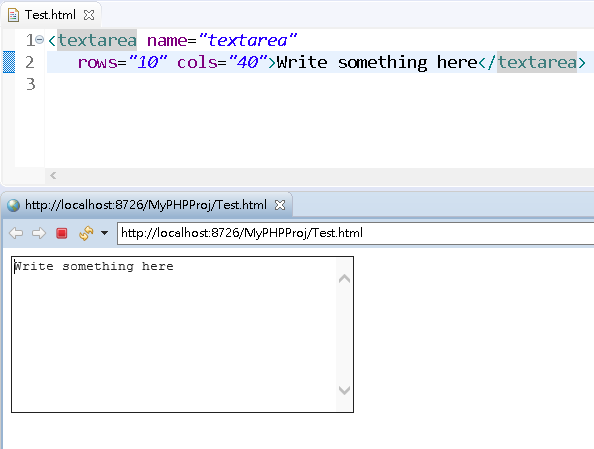
* The HTML <textarea> element represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.
  + cols屬性

The visible width of the text control, in average character widths.

If it is not specified, the default value is 20.

* + rows屬性

The number of visible text lines for the control.



### <button>

* The HTML <button> element represents a clickable button, which can be used in [forms](https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms) or anywhere in a document that needs simple, standard button functionality.
  + - type屬性值：

submit:

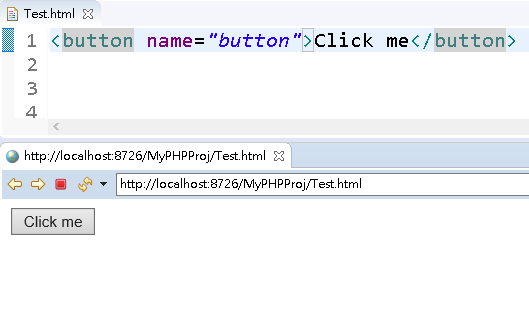
The button submits the form data to the server. This is the default if the attribute is not specified.

reset:

The button resets all the controls to their initial values.

button:

The button has no default behavior.



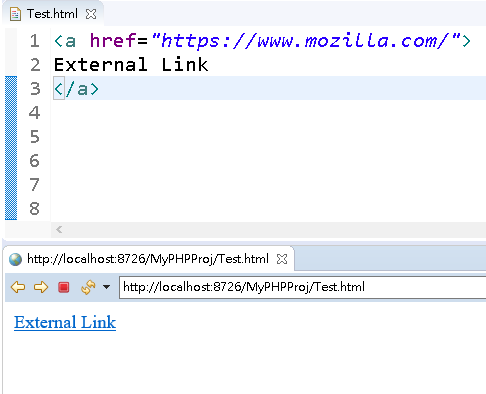
### <img>

* The HTML <img> element embeds an image into the document.



### <a>

* The HTML <a> element (or anchor element), along with its href attribute, creates a hyperlink to other web pages, files, locations within the same page, email addresses, or any other URL.



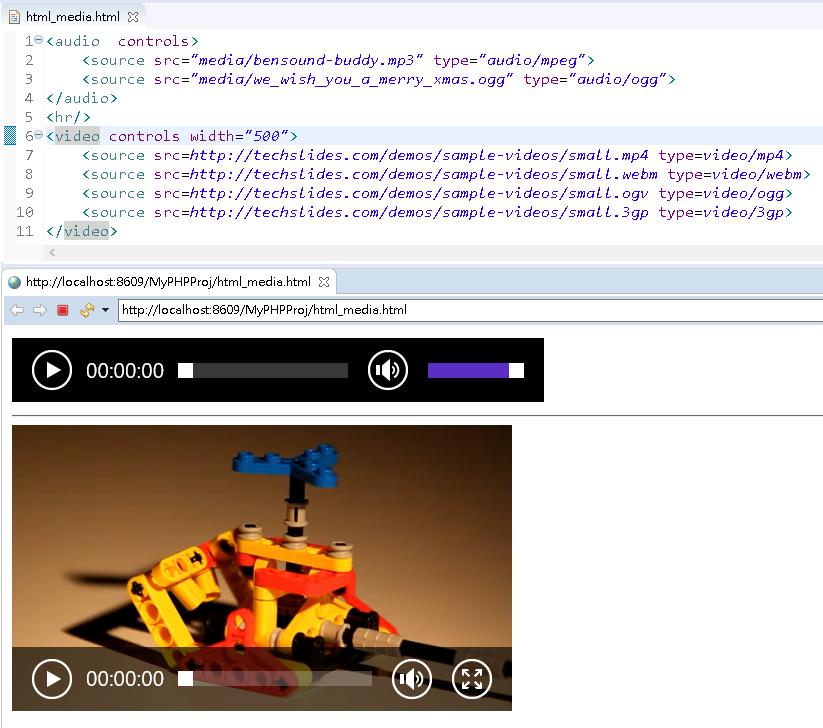
### <audio>

* + The HTML <audio> element is used to embed sound content in documents.
  + It may contain one or more audio sources, represented using the src attribute or the <source> element: the browser will choose the most suitable one.

### <video>

* + The HTML Video element (<video>) embeds a media player which supports video playback into the document.
  + You can use <video> for audio content as well, but the <audio> element may provide a more appropriate user experience

#### 範例：html\_media.html



### <table>表格相關元素

#### <table>

* + The HTML <table> element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.
    - border 屬性

This integer attribute defines, in pixels, the size of the frame surrounding the table.

#### <caption>

* The HTML Table Caption element (<caption>) specifies the caption (or title) of a table, and if used is always the first child of a [<table>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table).

#### <tr>

* The HTML <tr> element defines a row of cells in a table.
* The row's cells can then be established using a mix of <td> (data cell) and <th> (header cell) elements.

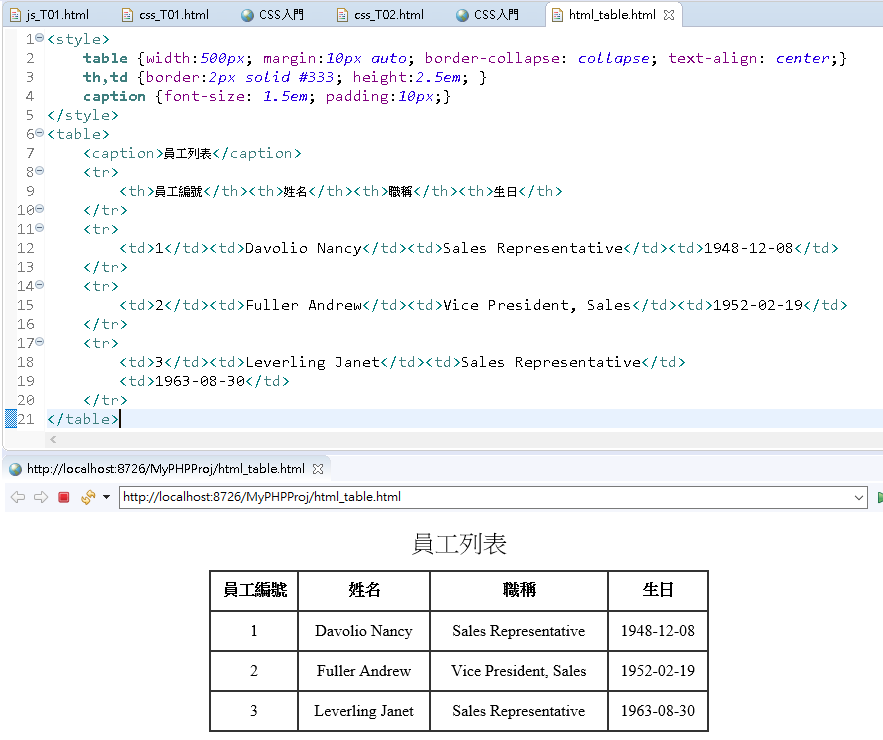
#### <th>

* The HTML <th> element defines a cell as header of a group of table cells.

#### <td>

* The HTML <td> element defines a cell of a table that contains data. It participates in the table model.

#### 範例I：html\_table.html

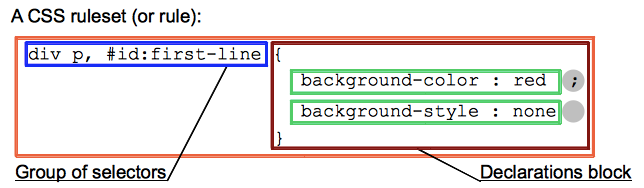


# CSS

## CSS簡介

* CSS(Cascading Style Sheet/串接樣式表/層疊樣式表)，是一種用於控制網頁樣式並允許樣式資訊與網頁內容分離的標記性語言。
* W3C於1996年審核通過CSS規格(CSS Level 1)。它以HTML為基礎，提供了豐富的格式化功能。目前W3C公告的最新建議規格為CSS Level 3 (2011年)。

## CSS樣式表規則

****

### 行內樣式表(In-line Style Sheets)

<h1 style="color:Red;font-size:2cm;">Hello Kitty</h1>

### 內嵌式樣式表(Embedded Style Sheets)

<style type="text/css">

h2 {color:Blue; font-size:2cm;}

</style>

### 外部連結式樣式表(External Style Sheets)

<link rel="Stylesheet" type="text/css" href="stylesheets/css\_T01.css" />

css\_T01.html

<!DOCTYPE html>

<html>

<head>

<title>CSS入門</title>

<link rel="Stylesheet" type="text/css" href="stylesheets/css\_T01.css" />

<!-- h3 { color:Green; font-size:2cm; } -->

<style type="text/css">

h2 {color:Blue; font-size:2cm;}

</style>

</head>body>

<h1 **style="color:Red;font-size:2cm;"**>Hello Kitty</h1>

<h2>Hello Mary</h2>

<h3>Hello Peter</h3>

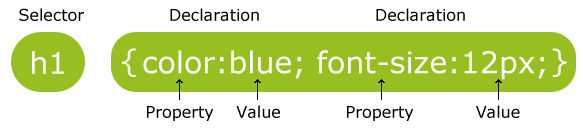
<h4>Hello Tom</h4>

</body>

</html>



## CSS選擇器



|  |  |  |  |
| --- | --- | --- | --- |
| **選擇器** | **範例** | **範例說明** | **CSS** |
| .class  (Class Selectors) | .intro | Selects all elements with class="intro" | 1 |
| #id  (ID Selectors) | #firstname | Selects the element with id="firstname" | 1 |
| element  (Type Selectors) | p | Selects all <p> elements | 1 |
| element,element  (Grouping Selectors) | div,p | Selects all <div> elements and all <p> elements | 1 |
| element element  (Descendant Selectors) | div p | Selects all <p> elements inside <div> elements | 1 |
| :link  (超連結假類別/link pseudo-class)表尚未被使用者點選的超連結 | a:link | Selects all unvisited links | 1 |
| :visited  (超連結假類別/link pseudo-class)表已被使用者點選後的超連結 | a:visited | Selects all visited links | 1 |
| :hover  (動態假類別/dynamic pseudo-class)表滑鼠指標正位於其上的元素 | a:hover | Selects links on mouse over | 1 |
| :active  (動態假類別/dynamic pseudo-class)表正被使用者**操作**的元素 | a:active | Selects the active link | 1 |
| :first-letter  (假元素pseudo-element) | p:first-letter | Selects the first letter of every <p> element | 1 |
| :first-line  (假元素pseudo-element) | p:first-line | Selects the first line of every <p> element | 1 |
| \* | \* | Selects all elements | 2 |
| element>element  (child selector) | div>p | Selects all <p> elements where the parent is a <div> element | 2 |
| element+element  (Adjacent sibling selector) | div+p | Selects all <p> elements that are placed immediately after <div> elements | 2 |
| [attribute]  (Attribute selector) | [target] | Selects all elements with a target attribute | 2 |
| [attribute=value] | [target=\_blank] | Selects all elements with target="\_blank" attribute | 2 |
| [attribute~=value] | [title~=flower] | Selects all elements with a title attribute containing the word "flower" | 2 |
| [attribute|=value] | [lang|=en] | Selects all elements with a lang attribute value starting with "en"  (如lang="en-abc"即為en開頭之lang屬性值) | 2 |
| :focus | input:focus | Selects the input element which has focus | 2 |
| :first-child | p:first-child | Selects every <p> element that is the first child of its parent | 2 |
| :before  (假元素pseudo-element) | p:before  如p:before {content: 'Hello'} | Insert content before  the content of every <p> element | 2 |
| :after  (假元素pseudo-element) | p:after  如p:after {content: 'Bye'} | Insert content after every <p> element | 2 |
| :lang(language) | p:lang(it) | Selects every <p> element with a lang attribute value starting with "it"  例如:  <p lang="it-xxx"> 或  <p lang="it"> | 2 |
| element1~element2  (General sibling selector) | p~ul | Selects every <ul> element that are preceded by a <p> element | 3 |
| [attribute^=value] | a[src^="https"] | Selects every <a> element whose src attribute value begins with "https" | 3 |
| [attribute$=value] | a[src$=".pdf"] | Selects every <a> element whose src attribute value ends with ".pdf" | 3 |
| [attribute\*=value] | a[src\*="w3schools"] | Selects every <a> element whose src attribute value contains the substring "w3schools" | 3 |
| :first-of-type | p:first-of-type | Selects every <p> element that is the first <p> element of its parent | 3 |
| :last-of-type | p:last-of-type | Selects every <p> element that is the last <p> element of its parent | 3 |
| :only-of-type | p:only-of-type | Selects every <p> element that is the only <p> element of its parent | 3 |
| :only-child | p:only-child | Selects every <p> element that is the only child of its parent | 3 |
| :nth-child(n) | p:nth-child(2) | Selects every <p> element that is the second child of its parent (n從1起算) | 3 |
| :nth-last-child(n) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child | 3 |
| :nth-of-type(n) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent | 3 |
| :nth-last-of-type(n) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child | 3 |
| :last-child | p:last-child | Selects every <p> element that is the last child of its parent | 3 |
| :root | :root | Selects the document’s root element | 3 |
| :empty | p:empty | Selects every <p> element that has no children (including text nodes) | 3 |
| :target | div:target | Selects the current active <div> element  <a>被點選時，其href所設定之「Id選擇器」所指向之目標元素  例如:  <style>  div:target {background-color:yellow; }  </style>  <div id="div1">AAA</div>  <div id="div2">BBB</div>  <a href="#div1">item1</a>  <a href="#div2">item2</a>  (點選item1超連結，則AAA所在之div背景顏色呈現黃色) | 3 |
| :enabled | input:enabled | Selects every enabled <input> element  (:enabled含<input>、<select>、<textarea>) | 3 |
| :disabled | input:disabled | Selects every disabled <input> element  (:disabled含<input>、<select>、<textarea>) | 3 |
| :checked | input:checked | Selects every checked <input> element  (指<input type='checkbox'>、<input type='radio'>) | 3 |
| :not(selector) | :not(p) | Selects every element that is not a <p> element  (註)  :not( ~~'p'~~ ) 🡪 selector前後不可加引號; jQuery則可 | 3 |
| ::selection  (假元素pseudo-element) | ::selection | Selects the portion of an element that is selected by a user | 3 |

css\_T02.html

<!DOCTYPE html>

<html>

<head>

<title>CSS入門</title>

<style type="text/css">

div.c1 {color:green; font-size:1cm;} /\*Class seletors\*/

div#div1 {color:red; font-size:1cm;} /\*ID selectors\*/

div {color:blue; font-size:1cm;} /\* Type selectors\*/

h1,h2 { color:lime; font-size:1cm; } /\* Grouping selectors\*/

body span { color:purple; font-size:1cm; } /\* Descendant selectors\*/

a:link {color:Blue; font-size:1cm; text-decoration:none}

a:visited {color:Green; font-size:1cm;}

a:hover {color:Red; font-size:1cm;}

a:active {color:Purple; font-size:1cm;}

</style>

</head>

<body>

<div class="c1">Hello Kitty</div>

<div id="div1">Hello Mary</div>

<div>Hello Peter</div>

<div>Hello Tom</div>

<h1>Hello Michael</h1>

<h2>Hello Hellen</h2>

<div>Hello <span>Carol</span></div>

<a href="#">資策會南區中心</a>

</body>

</html>



## CSS樣式屬性摘要

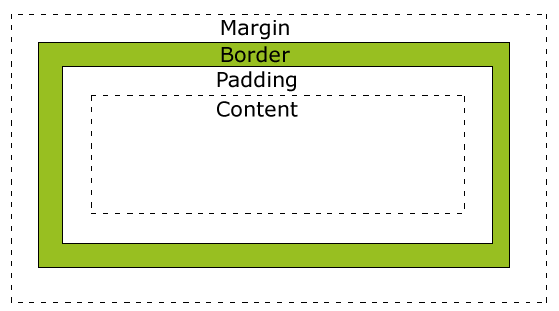
### 字型相關屬性

|  |  |
| --- | --- |
| **屬性名稱** | **屬性值** |
| font-style  (文字樣式) | * normal * italic (斜體) * oblique (斜體) |
| font-family  (字體) | 細明體, "Times New Roman" |
| font-variant  (文字變化) | * normal (標準) * small-caps (小型大寫字) |
| font-weight  (文字粗細) | * normal、bold、bolder、lighter * 100、200、300、**400**、500、600、**700**、800、900 * “normal”相當於是400，“bold”相當於是700 * “bolder”是指比**父元素**還要粗一級，“lighter”是指比**父元素**還要細一級 |
| font-size  (文字大小) | 1. 關鍵字:  * xx-small、x-small、small、medium、large、x-large、xx-large   font-size關鍵字之近似大小對照表   |  | | --- | | * 一般瀏覽器預設字型大小為16px; * <body>選擇器之預設字型大小為100%或1em |     資料來源：<http://www.trishasdesignstudio.com/font-size-conversion-chart.asp>   * larger、smaller (相較於**父元素**的文字大小)  1. 長度值:  * 數字 + 長度單位 * 常用長度單位  1. **em**：目前所使用之字型的大小。 2. **px**：pixels (1px = 1/96 in) 3. **cm**：公分 4. **mm**：公厘 5. **in**：inches (1in = 2.54cm = 96px) 6. **pt**：points (1pt = 1/72 in) 7. **pc**：picas (1pc = 12 pt = 1/6 in) 8. **ex**：目前所使用之字型的x-height大小(x-height通常為font-size的一半大小)。   http://www.myfirstfont.com/images/glyphterms.gif     1. %:   相較於**父元素**的文字大小的百分比 |
| line-height  (行距) | * normal * 數字 (目前font-size的 “數字”倍數) * % (目前font-size的百分比) * 長度值   **Leading** describes the amount of space between lines of text. You can measure leading by obtaining the distance between two baselines. Leading is called **line-height** in CSS.  http://cdn.designinstruct.com/files/243-basics_typography/03_leading.jpg |
| font  (多重屬性一併設定) | 依序設定 font-style font-variant font-weight font-size/line-height font-family  例如：font: italic normal bold 1cm/normal 標楷體;   * font-size、font-family必須設定 * 其餘屬性若省略，則代表設定預設值。   例如： **font**: 1cm 標楷體; |

### 文字相關屬性

|  |  |
| --- | --- |
| **屬性名稱** | **屬性值** |
| letter-spacing  (字元間距) | * normal (字元間無多餘間距) * 長度值 |
| text-decoration  (文字裝飾) | none、underline、overline、line-through、blink |
| text-align  (文字水平向  對齊) | left、right、center、justify |
| color  (文字顏色) | * rgb(255, 0, 0) 、 rgba(255, 0, 0, 0.5)   rgb(紅0~255,綠0~255,藍0~255) 、  rgba(紅,綠,藍,不透明度0~1)   * #ff0000 (十六進位數字) * red (關鍵字) |
| text-shadow  (文字陰影) | * text-shadow: 10px 10px 1px red;   text-shadow: h-shadow v-shadow blur color; |

### 盒子模型(Box Model)



來源：http://www.w3schools.com/css/css\_boxmodel.asp

|  |  |
| --- | --- |
| **屬性名稱** | ***屬性值*** |
| **width**  (content寬度) | * auto (瀏覽器按元素實際內容大小自行決定寬度) * % (相較於**外層**元素width的百分比) * 長度值 |
| **height**  (content高度) | * auto (瀏覽器按元素實際內容大小自行決定高度) * % (相較於**外層**元素height的百分比) * 長度值 |
| **padding**  padding-top  padding-bottom  padding-left  padding-right  (內距寬度) | * % (相較於**外層**元素width的百分比) * 長度值 (預設值0 )   **例如：**  **padding:10px 5px 15px 20px;** (上 右 下 左)  **padding:10px 5px 15px;** (上 左右 下)  **padding:10px 5px;** (上下 左右)  **padding:10px;** (四邊) |
| **border-width**  border-top-width  border-bottom-width  border-left-width  border-right-width  (邊框寬度) | * thin *(約1px)* * medium *(約2px)* * thick *(約4px)* * 長度值 |
| **border-style**  (邊框樣式) | none、dotted、dashed、solid、double、groove(溝)、ridge (脊)、inset(嵌入)、outset(突出) |
| **border-color**  (邊框顏色) | * 顏色值 * transparent |
| **border**  (多屬性一併設定) | * 依序設定border-style border-width border-color   例如：border: solid 1px red;  (註一)border允許只設定部分屬性;  (註二)三個屬性順序實不限; |
| **border-radius**  (邊框圓角) | * 長度值 (預設值0 )   border-radius: 10px; |
| **margin**  margin-top  margin-bottom  margin-left margin-right  (邊界寬度) | * auto (瀏覽器自行決定) * % (相較於**外層**元素width的百分比) * 長度值 (預設值0)   **例如：**  **margin:10px 5px 15px 20px;** (上 右 下 左)  **margin:10px 5px 15px;** (上 左右 下)  **margin:10px 5px;** (上下 左右)  **margin:10px;** (四邊) |
| **box-sizing**  (width與height  的計算模式) | * content-box (width、height只包含content一區) * border-box (width、height包含content、padding、border三區) |

### 背景相關屬性

|  |  |
| --- | --- |
| **屬性名稱** | ***屬性值*** |
| background-color  (背景顏色) | * transparent (透明的) * 顏色值 |
| background-image  (背景圖片) | * none * url(xxx.gif) |
| background-repeat  (背景圖片  重複方式) | * repeat-x * repeat-y * no-repeat * repeat |
| background-position  (背景圖片  起始位置) | * xpos ypos (網頁水平位置 網頁垂直位置) * x% y% (0% 0%) (相對於：背景區長/寬 減 背景圖長/寬) * left top   (**水平位置**可用關鍵字：**left、center、right**；  **垂直位置**可用關鍵字：**top、center、bottom**) |
| background  (多屬性一併設定) | 如:  body  { background: #ffffff url('hearts.gif') no-repeat  right top;} |

### Display(元素顯示)屬性

|  |  |
| --- | --- |
| **屬性名稱** | ***屬性值*** |
| display | * none * 隱藏元素 * block * 以垂直區塊顯示元素，如<h1>,<p>,<div> * 可設定區塊之width、height * inline * 以水平區塊顯示元素，如<span> * 無法設定區塊之width、height * inline-block * 以inline方式顯示元素，但實際上本質為block(可設定區塊之width、height) |

### 定位相關屬性

|  |  |
| --- | --- |
| **屬性名稱** | ***屬性值*** |
| float | * left * right * none * 利用float屬性可製作文繞圖的效果 |
| position | * static * absolute * 元素以絕對位置定位 (相對於position屬性值非static之最親近的祖先元素；若無此祖先元素，則相對於<html>元素) * 定位方法：使用left、top、right、bottom屬性 * fixed * 元素以絕對位置定位 (相對於瀏覽器視窗) * 定位方法：使用left、top、right、bottom屬性 * relative * 元素以相對位置定位 (相對於原本應在位置) * 位移方法：使用left、top、right、bottom屬性 |
| z-index | * auto * 整數 * z-index屬性用來調整重疊元素之上下位置 * z-index屬性值愈大者在上 |

### Cursor(滑鼠游標)屬性

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **CSS value** |  | **Description** |
| General | auto |  | The browser determines the cursor to display based on the current context. E.g. equivalent to text when hovering text. |
| default | default.gif | Default cursor, typically an arrow. |
| none |  | No cursor is rendered |
| Links & status | context-menu | context-menu.png | A context menu is available under the cursor. |
| help | help.gif | Indicating help is available. |
| pointer | pointer.gif | E.g. used when hovering over links, typically a hand. |
| progress | progress.gif | The program is busy in the background but the user can still interact with the interface (unlike for wait). |
| wait | wait.gif | The program is busy (sometimes an hourglass or a watch). |
| Selection | cell | cell.gif | Indicating that cells can be selected. |
| crosshair | crosshair.gif | Cross cursor, often used to indicate selection in a bitmap. |
| text | text.gif | Indicating text can be selected, typically an I-beam. |
| vertical-text | vertical-text.gif | Indicating that vertical text can be selected, typically a sideways I-beam. |
| Drag and drop | alias | alias.gif | Indicating an alias or shortcut is to be created. |
| copy | copy.gif | Indicating that something can be copied. |
| move | move.gif | The hovered object may be moved. |
| no-drop | no-drop.gifCursor showing that a drop is not allowed at the current location. | |
| not-allowed | not-allowed.gif | Cursor showing that something cannot be done. |
| Resize & scrolling | all-scroll | all-scroll.gif | Cursor showing that something can be scrolled in any direction (panned). |
| col-resize | col-resize.gif | The item/column can be resized horizontally. Often rendered as arrows pointing left and right with a vertical bar separating. |
| row-resize | row-resize.gif | The item/row can be resized vertically. Often rendered as arrows pointing up and down with a horizontal bar separating them. |
| n-resize | Example of a resize towards the top cursor | Some edge is to be moved. For example, the se-resize cursor is used when the movement starts from the *south-east* corner of the box. |
| e-resize | Example of a resize towards the right cursor |
| s-resize | Example of a resize towards the bottom cursor |
| w-resize | Example of a resize towards the left cursor |
| ne-resize | Example of a resize towards the top-right corner cursor |
| nw-resize | Example of a resize towards the top-left corner cursor |
| se-resize | Example of a resize towards the bottom-right corner cursor |
| sw-resize | Example of a resize towards the bottom-left corner cursor |
| ew-resize | 3-resize.gif | Indicates a bidirectional resize cursor. |
| ns-resize | 6-resize.gif |
| nesw-resize | 1-resize.gif |
| nwse-resize | 4-resize.gif |
| Zoom | zoom-in | zoom-in.gif | Indicates that something can be zoomed (magnified) in or out. |
|  |  |

## CSS選擇器優先順序

### 選擇器具體性(selector's specificity)計分規則

|  |  |  |
| --- | --- | --- |
| **順序** | **規則** | **結果** |
| 1 | style="樣式表" | a=1, b=0, c=0, d=0 |
| 2 | Id選擇器 | a=0, b=1, c=0, d=0 |
| 3 | 其他**屬性或假類別**選擇器 | a=0, b=0, c=1, d=0 |
| 4 | **元素或假元素**選擇器 | a=0, b=0, c=0, d=1 |

### 選擇器具體性計分範例

|  |  |  |
| --- | --- | --- |
| 1 | \* {} | a=0 b=0 c=0 d=0 |
| 2 | li {} | a=0 b=0 c=0 d=1 |
| 3 | li:first-line {} | a=0 b=0 c=0 d=2 |
| 4 | ul li {} | a=0 b=0 c=0 d=2 |
| 5 | ul ol+li {} | a=0 b=0 c=0 d=3 |
| 6 | h1 + \*[rel=up] {} | a=0 b=0 c=1 d=1 |
| 7 | ul ol li.red {} | a=0 b=0 c=1 d=3 |
| 8 | li.red.level {} | a=0 b=0 c=2 d=1 |
| 9 | #x34y {} | a=0 b=1 c=0 d=0 |
| 10 | style="" | a=1 b=0 c=0 d=0 |

(實例1)

<html>

<head>

<title>CSS Specificity</title>

<style type="text/css">

li.red.level {color:red;}

ul ol li.red { color:green;}

</style>

</head>

<body style="font-size:1cm;">

<ul>

<li>Programming Language

<ol>

<li class="red level">JavaScript</li>

<li>jQuery</li>

</ol>

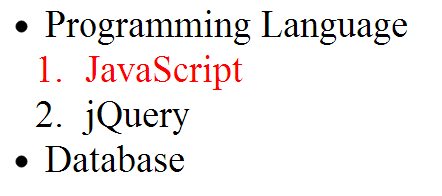
</li>

<li>Database</li>

</ul>

</body>

</html>



(實例2)

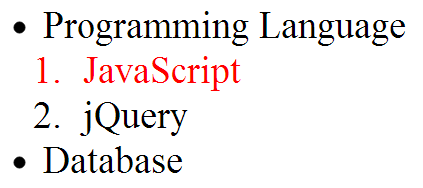
<style type="text/css">

ul [class].red { color:green;}

li.red.level {color:red;}

|  |
| --- |
| 優先權相同時，以位置在**後**者為準 |

</style>



# JavaScript

## JavaScript基本語法

* JavaScript程式內嵌於HTML網頁裡，由敘述(statement)所組成並以<script>標籤區塊包圍起來。
* 一個敘述相當於一道電腦指令，每個敘述不需使用分號(；)結尾，但多個敘述放在同一列時，敘述與敘述之間需以分號(；)間隔。
* JavaScript的註解：
  + 單行註解(inline comment) //
  + 區塊註解(block comment) /\* \*/
* 寫一支最簡單的JavaScript程式

js\_T01.html

<html>

<head>

<title>JavaScript入門</title>

</head>

<body>

<h1>Hello

**<script type="text/javascript">**

//document:網頁物件(JavaScript內建物件之一)

**document.write("JavaScript")**

**</script>**

</h1>

</body>

</html>

|  |
| --- |
| Hello JavaScript |

## 變數

### 宣告變數

* 變數是暫存資料的一塊記憶體空間。
* JavaScript變數使用前需先宣告，但無需指定變數的datatype

var x;   
var carname;

* 宣告+初始化變數

var x=5;  
var carname="Volvo";

* 變數只宣告未初始化，變數值為undefined

### 變數名稱基本規則

* 字母、底線\_、錢字$符號開頭
* 第二個字元起可為字母、底線\_、錢字$符號或數字
* 區分大小寫
* 不能使用關鍵字/保留字

### 變數自動宣告

* 當變數直接指定一個值之前未使用var宣告時，JavaScript會自動(implicitly)為你宣告該變數 且宣告為全域變數
* 在函數內宣告的變數為區域變數(local variable)
* 函數外宣告的變數為全域變數(global variable)

x=5  
carname="Volvo"

**//以上敘述相當於如下敘述**

var x

x=5

var carname

carname="Volvo"

## JavaScript支援的常用資料型別

### 原始資料型別(Primitive Data Type)

#### 字串型別(String)

* 字串型別代表一種字元序列，通常用來表示一組文字資料
* 字串內每個字元對應16-bit之正整數
* 字串型別之值須以**單引號**或**雙引號**括起來，如 "Hello" 或 'Hello'

#### 數值型別(Number)

* + Number型別之值只有64位元浮點數 (包含整數)，例如：

var n1=10;

var n2=10.5;

#### 布林型別(Boolean)

* 布林型別之值只有true 、false

#### Null型別

* Null型別之值只有null
* null表示不存在的物件

#### 未定義型別(Undefined)

* 未定義型別之值只有undefined
* 一個被宣告而沒有初始化的變數有undefined值

### 物件(Object)

* 物件是一個包含資料與處理資料指令的資料結構。
* Object refers to a data structure containing data and instructions for working with the data.

## 運算子

### 算術運算子

var y=5;

|  |  |  |  |
| --- | --- | --- | --- |
| **運算子** | **說明** | **範例** | **結果** |
| + | Addition | x=y+2 | x=7 |
| - | Subtraction | x=y-2 | x=3 |
| \* | Multiplication | x=y\*2 | x=10 |
| / | Division | x=y/2 | x=2.5 |
| % | Modulus (division remainder) | x=y%2 | x=1 |
| ++ | Increment | x=++y | x=6 |
| -- | Decrement | x=--y | x=4 |

### 字串串連運算子

x = 5 + "5";  
document.write(x); **//"55"**

### 比較運算子

**var x=5**

|  |  |  |
| --- | --- | --- |
| **運算子** | **說明** | **範例** |
| == | is equal to | x==5 is true  x=='5' is true |
| === | is exactly equal to (value and type) | x===5 is true x==='5' is false |
| != | is not equal | x != '5' is false |
| !== | is not exactly equal to | x !=='5' is true |
| > | is greater than | x>8 is false |
| < | is less than | x<8 is true |
| >= | is greater than or equal to | x>=8 is false |
| <= | is less than or equal to | x<=8 is true |

### 邏輯運算子

**x=6; y=3;**

|  |  |  |
| --- | --- | --- |
| **運算子** | **說明** | **範例** |
| && | and | (x < 10 && y > 1) is true |
| || | or | (x==5 || y==5) is false |
| ! | not | !(x==y) is true |

### 條件運算子

var lang = "CHINESE"

var greeting= (lang=="CHINESE")?"歡迎":"Welcome";

**//當lang=="CHINESE"為真，則greeting被指定"歡迎"，否則greeting被指定"Welcome "**

### 指定運算子

**x=10;**  **y=5;**

|  |  |  |  |
| --- | --- | --- | --- |
| **運算子** | **說明** | **Same As** | **結果** |
| = | x=y |  | x=5 |
| += | x+=y | x=x+y | x=15 |
| -= | x-=y | x=x-y | x=5 |
| \*= | x\*=y | x=x\*y | x=50 |
| /= | x/=y | x=x/y | x=2 |
| %= | x%=y | x=x%y | x=0 |

## 運算子優先順序

|  |  |
| --- | --- |
| 運算子 | 說明 |
| .  []  () | Field access  Array indexing  Function calls |
| ++  --  - (負)  !  typeof (回傳data type) | Unary operators |
| \* / % | Multiplication, division, modulo division |
| + - + | Addition, subtraction, string concatenation |
| < ,<=  >, >= | Less than, less than or equal  Greater than, greater than or equal |
| ==, !=  ===, !== | Equality, inequality  Strict equality, strict inequality |
| && | Logical AND |
| || | Logical OR |
| ?: | Conditional |
| =  *OP*= | Assignment  Assignment with operation (such as += and &=) |

## 流程控制結構

### 選擇性結構

#### if 敘述

if (條件式/condition)  
  {  
  程式碼被執行(當條件式為true)  
  }

* condition為0、''、null、undefined時，視為false；
* condition為其他值時，則視為true。

<script type="text/javascript">

var gender="Male";

if (gender == "Male") {

document.write("男性網友您好!");

}

</script>

**//結果：**

**男性網友您好!**

#### if...else敘述

if ((條件式/condition)  
  {  
  程式碼被執行(當條件式為true)

}  
 else  
  {  
  程式碼被執行(當條件式為false)  
  }

<script type="text/javascript">

//prompt(text [,defaultText] ) : 輸入對話框

var gender = prompt("請輸入性別: 1(男) 2(女)" , "1");

if (gender == "1") {

document.write("男性網友您好!");

}

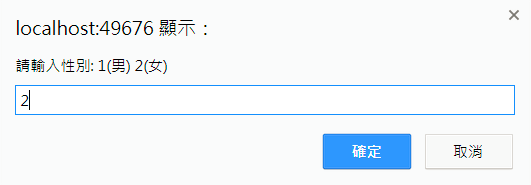
else {

document.write("女性網友您好!");

}

</script>

**//當輸入對話框填入2**

****

**結果：**

**女性網友您好!**

#### switch敘述

switch(n)  
{  
 case 1:  
  execute code block 1  
  break;  
 case 2:  
  execute code block 2  
  break;  
 default:  
  code to be executed if n is different from case 1 and 2  
}

<script type="text/javascript">

var birthPlace="台北";

switch (birthPlace)

{

case "台南":

document.write("出生地:台南");

break;

case "台中":

document.write("出生地:台中");

break;

case "台北":

document.write("出生地:台北");

break;

default:

document.write("出生地:N/A");

break;

}

</script>

**//結果：**

**出生地:台北**

### 重複性結構

#### for loop (for迴圈)

for ([var] v=起值**;** v<=終值**;** v=v+增值)  
{  
 程式碼被執行  
}

<script type="text/javascript">

var total = 0;

for (var i = 1; i <= 10; i++) {

total += i;

document.write("i=" + i + " total=" + total+"<br/>");

}

</script>

**//結果：**

**i=1 total=1  
i=2 total=3  
i=3 total=6  
i=4 total=10  
i=5 total=15  
i=6 total=21  
i=7 total=28  
i=8 total=36  
i=9 total=45  
i=10 total=55**

#### while loop (while迴圈)

while (條件式/condition)  
 {  
  程式碼被執行(當條件式為true)  
 }

#### do while loop

do  
 {  
  程式碼被執行  
 }

while (條件式/condition);

#### for...In loop

for ([var] index in array)  
 {  
  程式碼被執行  
 }

for ([var] property in object)  
 {  
  程式碼被執行  
 }

## 陣列

### 建立陣列

#### 使用Array建構方法

var myCars=**new Array();**  //new Array(5)亦可  
myCars[0]="Saab";

myCars[1]="Volvo";

myCars[2]="BMW";

#### 使用Array Literal

var myCars=["Saab","Volvo","BMW"];

### 存取陣列元素

#### 存取特定位置之陣列元素

var myCars = ["Saab","Volvo","BMW"];

document.write(myCars.length + "<br/>"); //結果：3

myCars[10] = "Rolls";

document.write(myCars.length); //結果：11

document.write("<h2>" + myCars.toString() + "</h2>");

//結果：Saab,Volvo,BMW,,,,,,,,Rolls

#### 使用for loop巡覽每一個陣列元素

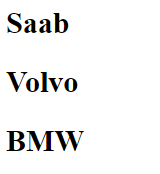
var myCars = ["Saab", "Volvo", "BMW"];

for (var i = 0; i < **myCars.length**; i++) {

document.write("<h2>" + myCars[i] + "</h2>");

}

//結果：



#### 使用for in loop巡覽每一個陣列元素

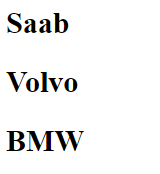
var myCars = ["Saab", "Volvo", "BMW"];

for (var index in myCars) {

document.write("<h2>"+ myCars[index] + "</h2>");

}

//結果：



## 函數

### 一般函數

#### 函數定義

function functionname(var1,var2,...,varX)  
{  
 some code  
}

#### 函數範例

js\_T02.html

<!DOCTYPE html>

<html>

<head>

<title>JavaScript入門</title>

<style type="text/css">

div { width:500px; margin:50px auto; font-size:1cm; }

input { font-size:1cm; }

</style>

<script type="text/javascript">

//函數定義

function sayHello() {

var name=document.getElementById('text1').value;

alert("歡迎"+name+"光臨...");

}

</script>

</head>

<body>

<div>

姓名<input type="text" id="text1" size="10" />

<input type="button" value="確定" onclick="sayHello();//函數呼叫" />

</div>

</body>

</html>



### function literal(無名函數)

#### 建立函數

*function(參數1,參數2,…) { 函數body }*

#### 函數範例

<html>

<body>

<script type="text/javascript">

var f1 = function(x,y){return Math.pow(x,y);};

alert( f1(2,3) ); //顯示8

f2=f1;

alert( f2(3,2) ); //顯示9

function browseArray(callback) {

var nums = [11, 12, 13, 14, 15];

for (var i = 0; i < nums.length; i++) {

callback(i, nums[i]);

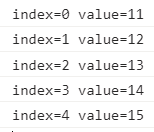
}

}

browseArray(function (index, value) {

console.log("index=%d value=%s", index, value);

});



</script>

</body>

</html>

## 物件深入探析

### 建立物件

#### 使用JS內建Object建構子函數(Constructor)

var obj1 = new Object();

obj1.name = "Mary";

obj1.age = 1;

obj1.cry = function () {

return this.name+"哭哭哭...";

}

#### 使用Object Literal語法

var obj2 = {

"name": "Peter",

"age": 2,

"cry": function () {

return this.name+"哇哇哇...";

}

}

#### 使用使用者自訂建構子函數(Constructor)

function Cat() {

this.name = "Kitty";

this.age = 3;

this.cry = function () {

return this.name+"喵喵喵...";

}

}

var cat1 = new Cat();

### 使用物件

* 使用「物件變數.property名稱」存取物件內之屬性
* 使用「物件變數.method名稱( )」呼叫物件內之方法

js\_T03.html

<!DOCTYPE html>

<html>

<head>

<title>JavaScript入門</title>

<link rel="icon" href="../favicon.ico" />

</head>

<body>

<script type="text/javascript">

var obj1 = new Object();

obj1.name = "Mary";

obj1.age = 1;

obj1.cry = function () {

return this.name+"哭哭哭...";

}

document.write("<h2> 姓名: " + obj1.name + " 年齡: " + obj1.age +

"歲 " + obj1.cry() + "</h2>");



var obj2 = {

"name": "Peter",

"age": 2,

"cry": function () {

return this.name+"哇哇哇...";

}

}

document.write("<h2> 姓名: " + obj2.name + " 年齡: " + obj2.age +

"歲 " + obj2.cry() + "</h2>");



function Cat(name,age) {

this.name = name;

this.age = age;

this.cry = function () {

return this.name + "喵喵喵...";

}

}

var cat1 = new Cat("Kitty",3);

document.write("<h2> 姓名: " + cat1.name + " 年齡: " + cat1.age +

"歲 " + cat1.cry() + "</h2>");



# jQuery

## jQuery 簡介

* jQuery是一套跨瀏覽器的JavaScript函式庫(Function Library)，強化了HTML與JavaScript之間的操作，是目前最受歡迎的JavaScript函式庫。
* 依據W3Techs於2018年5月對全球人氣排名前一千萬大網站所作的統計報告：
  + 最多網站所使用的用戶端(Client-Side)程式語言：

|  |  |  |
| --- | --- | --- |
| 程式語言 | | 使用率 |
| 1. | [JavaScript](https://w3techs.com/technologies/details/cp-javascript/all/all) | 94.9% |
| 2. | [Flash](https://w3techs.com/technologies/details/cp-flash/all/all) | 4.7% |
| 3. | [Silverlight](https://w3techs.com/technologies/details/cp-silverlight/all/all) | 0.1% |

* + 最多網站所使用的JavaScript 函式庫：

|  |  |  |
| --- | --- | --- |
| 程式語言 | | 使用率 |
| 1. | [jQuery](https://w3techs.com/technologies/details/js-jquery/all/all) | 73.2% |
| 2. | [Bootstrap](https://w3techs.com/technologies/details/js-bootstrap/all/all) | 17.3% |

* jQuery的第一個版本是由John Resig在2006年1月的BarCamp NYC上所釋出的。 目前則由成立於2009年9月份的 「The jQuery Project」 (http://jquery.com/) 繼續研發，不定期推出更新版本。

## jQuery選擇器

### jQuery選擇器列表

|  |  |  |
| --- | --- | --- |
| 選擇器 | 範例 | 說明 |
| \* | $("\*") | All elements |
| #*id* | $("#lastname") | The element with id="lastname" |
| .*class* | $(".intro") | All elements with class="intro" |
| .*class,*.*class* | $(".intro,.demo") | All elements with the class "intro" or "demo" |
| *element* | $("p") | All <p> elements |
| *el1*,*el2*,*el3* | $("h1,div,p") | All <h1>, <div> and <p> elements |
|  |  |  |
| :first (jQuery自訂) | $("p:first") | The first <p> element |
| :last | $("p:last") | The last <p> element |
| :even | $("tr:even") | All even <tr> elements |
| :odd | $("tr:odd") | All odd <tr> elements |
|  |  |  |
| :first-child | $("p:first-child") | All <p> elements that are the first child of their parent |
| :first-of-type | $("p:first-of-type") | All <p> elements that are the first <p> element of their parent |
| :last-child | $("p:last-child") | All <p> elements that are the last child of their parent |
| :last-of-type | $("p:last-of-type") | All <p> elements that are the last <p> element of their parent |
| :nth-child(*n*) | $("p:nth-child(2)") | All <p> elements that are the 2nd child of their parent (index starts at 1) |
| :nth-last-child(*n*) | $("p:nth-last-child(2)") | All <p> elements that are the 2nd child of their parent, counting from the last child |
| :nth-of-type(*n*) | $("p:nth-of-type(2)") | All <p> elements that are the 2nd <p> element of their parent |
| :nth-last-of-type(*n*) | $("p:nth-last-of-type(2)") | All <p> elements that are the 2nd <p> element of their parent, counting from the last child |
| :only-child | $("p:only-child") | All <p> elements that are the only child of their parent |
| :only-of-type | $("p:only-of-type") | All <p> elements that are the only child, of its type, of their parent |
|  |  |  |
| parent > child | $("div > p") | All <p> elements that are a direct child of a <div> element |
| parent descendant | $("div p") | All <p> elements that are descendants of a <div> element |
| element + next | $("div + p") | The <p> element that are next to each <div> elements  (Adjacent sibling) |
| element ~ siblings | $("div ~ p") | All <p> elements that are siblings of a <div> element (General sibling) |
|  |  |  |
| :eq(*index*) | $("ul li:eq(3)") | The fourth element in a list (index starts at 0) |
| :gt(*no*) | $("ul li:gt(3)") | List elements with an index greater than 3 |
| :lt(*no*) | $("ul li:lt(3)") | List elements with an index less than 3 |
| :not(*selector*) | $("input:not([type=checkbox])") | All input elements that are not checkbox |
|  |  |  |
| :header | $(":header") | All header elements <h1>, <h2> ... |
| :animated | $(":animated") | All animated elements |
| :focus | $(":focus") | The element that currently has focus  例如  <body> <input type="text" /></body>  <script>  $(function () {  $("input:eq(0)").focus();  $(":focus").css({color:'red'})  })  </script> |
| :contains(*text*) | $(":contains('Hello')")  註：  1. 引號'可省略。  2. 關鍵字(Hello)前後不可有多餘的空白 | All elements which contains the text "Hello" |
| :has(*selector*) | $("div:has(p)") | All <div> elements that have a 子孫<p> element |
| :empty | $(":empty") | All elements that are empty |
| :parent | $(":parent") | All elements that are a parent of another element (或文字節點) |
| :hidden | $("p:hidden") | All hidden <p> elements (包含<head>區塊內) |
| :visible | $("table:visible") | All visible tables |
| :root | $(":root") | The document’s root element  (即<html>) |
| :lang(*language*) | $("p:lang(zh)") | All <p> elements with a lang attribute value starting with "zh" (指lang="zh-xxx") |
|  |  |  |
| [*attribute*] | $("[href]") | All elements with a href attribute |
| [*attribute*=*value*] | $("[href='default.htm']") | All elements with a href attribute value equal to "default.htm"(value如含特殊字元，如.或空白，須以引號括起來) |
| [*attribute*!=*value*] | $("[href!='default.htm']") | All elements with a href attribute value not equal to "default.htm" |
| [*attribute*$=*value*] | $("[href$='.jpg']") | All elements with a href attribute value ending with ".jpg" |
| [*attribute*|=*value*] | $("[hreflang|='en']") | All elements with a hreflang attribute value starting with "en" (以en開頭指en後面接 -xxx之字串) |
| [*attribute*^=*value*] | $("[name^='hello']") | All elements with a name attribute value starting with "hello"(以hello開頭指hello後面接任意字串) |
| [*attribute*~=*value*] | $("[name~='hello']") | All elements with a name attribute value containing the word "hello" (word指以空白字元間隔之字串) |
| [*attribute\**=*value*] | $("[name\*='hello']") | All elements with a name attribute value containing the string "hello" |
|  |  |  |
| :input | $(":input") | All input elements  (含<select>、<textarea>) |
| :text | $(":text") | All input elements with type="text" |
| :password | $(":password") | All input elements with type="password" |
| :radio | $(":radio") | All input elements with type="radio" |
| :checkbox | $(":checkbox") | All input elements with type="checkbox" |
| :submit | $(":submit") | All input elements with type="submit"或 <button type='submit'></button> |
| :reset | $(":reset") | All input elements with type="reset"或 <button type='reset'></button> |
| :button | $(":button") | All input elements with type="button"或 <button type= 'button | submit | reset'></button> |
| :image | $(":image") | All input elements with type="image" |
| :file | $(":file") | All input elements with type="file" |
| :enabled | $(":enabled") | All enabled input elements (含<select>、<textarea>、<fieldset>) |
| :disabled | $(":disabled") | All disabled input elements |
| :selected | $(":selected") | All selected <option> elements |
| :checked | $(":checked") | All checked input elements(含<input type='checkbox | radio'>、<select><option></option></select> ) |

### jQuery選擇器使用簡例(jq\_T01.html)

jq\_T01.html

<!DOCTYPE html>

<html>

<head>

<title>jQuery入門</title>

<script type="text/javascript"

src="javascripts/jquery-3.4.1.min.js"></script>

<!-- <script type="text/javascript"

src="https://code.jquery.com/jquery-3.4.1.min.js"></script> -->

<script type="text/javascript">

function f1() {

/\* 類別名稱c1之<div>內的<h1>的文字顏色設定為紅色 \*/

//jQuery API

$("div.c1 h1").css("color", "red" );

**/\*** DOM API

var h1s = document.getElementsByTagName('h1');

for (i = 0; i < h1s.length; i++) {

if(h1s[i].parentNode.nodeName=='DIV' &&

h1s[i].parentNode.className=='c1')

h1s[i].style.color = 'red';

}

\*/

}

</script>

</head>

<body onload="f1();">

<div class="c1">

<h1>Hello Kitty</h1>

<h1>Hello Mary</h1>

<h1>Hello Peter</h1>

</div>

<div class="c2">

<h1>Hello Kitty</h1>

<h1>Hello Mary</h1>

<h1>Hello Peter</h1>

</div>

</body>

</html>



## jQuery常用API

### 存取元素之CSS樣式

#### .css( propertyName, value ) Returns: jQuery

* + Description: Set one or more CSS properties for the set of matched elements.

.css( propertyName, value )

**propertyName**

Type: String

A CSS property name.

**value**

Type: String or Number

A value to set for the property.

.css( properties )

**properties**

Type: PlainObject

An object of property-value pairs to set.

#### .addClass( className ) Returns: jQuery

* + Description: Adds the specified class(es) to each element in the set of matched elements.

.addClass( className )

className

Type: String

One or more space-separated classes to be added to the class attribute of each matched element.

#### .removeClass( [className ]) Returns: jQuery

* + Description: Remove a single class, multiple classes, or all classes from each element in the set of matched elements.

.removeClass( [className ] )

className

Type: String

One or more space-separated classes to be removed from the class attribute of each matched element.

#### 範例：jq\_T02.html

jq\_T02.html

<html>

<head>

<title>jQuery入門</title>

<style type="text/css">

.style1 {color:Blue; font-size:1cm; font-weight:bold;}

.style2 {font-style:italic;}

</style>

<script type="text/javascript"

src="javascripts/jquery-3.4.1.min.js"></script>

<script type="text/javascript">

function changeStyle() {

$("#div1").css('font-style','italic').css('color','blue')

.css('font-size','1cm').css('font-weight','bold');

}

function resetStyle() {

$("#div1").css({ 'font-style':'', color: '', 'font-size': '',

'font-weight': '' });

//使用 "Object literal"當引數，可一次設定多項樣式屬性

//Object literal結構: {property:value,property:value}

}

function changeStyle2() {

$("#div2").addClass("style1 style2");

}

function resetStyle2() {

$("#div2").removeClass("style1 style2");

}

</script>

</head>

<body>

<div id="div1" onmouseover="changeStyle()"

onmouseout="resetStyle()">

Hello Kitty<br />小甜甜</div>

<hr />

<div id="div2" onmouseover="changeStyle2()"

onmouseout="resetStyle2()">

Hello Kitty<br />小甜甜</div>

</body>

</html>



### 存取元素之屬性

#### .attr( attributeName ) Returns: String

* + Description: Get the value of an attribute for the first element in the set of matched elements.

.attr( attributeName )

attributeName

Type: String

The name of the attribute to get.

#### .attr(attributeName, value) Returns: jQuery

* + Description: Set one or more attributes for the set of matched elements.

.attr( attributeName, value )

**attributeName**

Type: String

The name of the attribute to set.

**value**

Type: String or Number or Null

A value to set for the attribute. If null, the specified attribute will be removed.

.attr( attributes )

**attributes**

Type: PlainObject

An object of attribute-value pairs to set.

#### 範例：jq\_T03.html(page I)

jq\_T03.html(第一頁)

<html>

<head>

<title>jQuery入門</title>

<link rel="stylesheet" type="text/css"

href="plugins/ jQuery UI 1.12.1/stylesheets/base/jquery-ui.css" />

<script type="text/javascript"

src="javascripts/jquery-3.4.1.min.js"></script>

<script type="text/javascript"

src="plugins/jQuery UI 1.12.1/jquery-ui.min.js"></script>

<script type="text/javascript">

function init() {

$("#tabs").tabs();//將網頁改造成「多標籤頁」模式顯現

f1();//存取特定節點之屬性

}

function f1(){

//求取特定元素之屬性

var s1 = $("h1:eq(1)").attr('title');

//設定特定元素之屬性值

$("h1:eq(1)").attr({ title: s1+'超Cute!' });

}

</script>

</head>

<body onload="init()">

<div id="tabs">

<ul>

<li><a href="#p1">存取特定節點之屬性</a></li>

</ul>

<div id="p1">

<h1 title="粉可愛"!>Hello Kitty</h1>

<h1 title="超可愛!">小甜甜</h1>

</div>

</div>

</body>

</html>



### 巡覽jQuery集合物件內的每一元素

#### .attr( attributeName ) Returns: String

* + Description: Get the value of an attribute for the first element in the set of matched elements.

.attr( attributeName )

attributeName

Type: String

The name of the attribute to get.

#### .each( function ) Returns: jQuery

* + Description: Iterate over a jQuery object, executing a function for each matched element.

.each( function )

function

Type: Function( Integer **index**, Element **element** )

A function to execute for each matched element.

#### .html( ) Returns: String

* + Description: Get the HTML contents of the first element in the set of matched elements.

.html()

This method does not accept any arguments.

#### .html( htmlString ) Returns: jQuery

* + Description: Set the HTML contents of each element in the set of matched elements.

.html( htmlString )

htmlString

Type: htmlString

A string of HTML to set as the content of each matched element.

#### 範例：jq\_T03.html(page II)

jq\_T03.html (第二頁)

<!DOCTYPE html>

<html>

<head>

(略)

<script type="text/javascript">

$(function () {

$("#tabs").tabs();

});

function init() {

f2(); //巡覽jQuery集合物件內的每一元素

}

function f2() {

var s = "";

$("#p1 h1").each(function () {

s += "<h3>" + $(this).html() + "</h3>";

//取元素內容

});

$("#p2").html(s);//設定元素內容

}

</script>

</head>

<body onload="init()">

<div id="tabs">

<ul>

<li><a href="#p1">存取特定節點之屬性</a></li>

<li><a href="#p2">巡覽jQuery集合物件內的每一元素</a></li>

</ul>

<div id="p1">

<h1 title="粉可愛"!>Hello Kitty</h1>

<h1 title="超可愛!">小甜甜</h1>

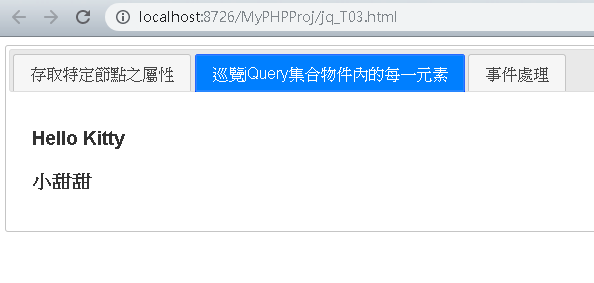
</div>

<div id="p2"></div>

</div>

</body>

</html>



### 事件處理

#### .ready( handler ) Returns: jQuery

* + Description: Specify a function to execute when the DOM is fully loaded.

.ready( handler )

**handler**

Type: Function()

A function to execute after the DOM is ready.

* + jQuery offers several ways to attach a function that will run when the DOM is ready. All of the following syntaxes are equivalent:

$( handler )

$( document ).ready( handler )

$().ready( handler )

#### .click( handler ) Returns: jQuery

* + Description: Bind an event handler to the "click" JavaScript event.

.click( handler )

**handler**

Type: Function( Event eventObject )

A function to execute each time the event is triggered.

#### 範例：jq\_T03.html(page III)

jq\_T03.html (第三頁)

<!DOCTYPE html>

<html>

<head>

(略)

<script type="text/javascript">

/\*

//window.onload為當網頁內容被完全載入(事件)後，觸發執行的event handler

window.onload = function () {

document.getElementById("button1").onclick =

function () {

alert("歡迎"+

document.getElementById("text1").value+"光臨!");

}

};

\*/

$(document).ready( **function**(){

$("#p3 #button1").click(

**function**() { alert("歡迎" + $("#text1").val() + "光臨!"); } );

});

/\*

1.$(document).ready( functionName );

用於取代window.onload,且可呼叫多次而不會互相衝突

2.“ready事件”觸發於DOM完成建立後;

“load事件”觸發於網頁相關檔案都已下載後

\*/

</script>

</head>

<body>

<div id="tabs">

<ul>

<li><a href="#p3">事件處理</a></li>

</ul>

<div id="p3">

<div style="width:*600px*; margin:*30px auto*; padding:*30px*;

border:*1px solid #333*; text-align:*center*;">

姓名 <input type=*"text"* id=*"text1"* size=*"20"* />

<input type=*"button"* id=*"button1"* value=*"Click"* />

</div>

</div>

</div>

</body>

</html>



### 特效與動畫

#### .show() Returns: jQuery

* + Description: Display the matched elements.

.show()

This signature does not accept any arguments.

.show( [duration ] [, complete ] )

**duration (default: 400)**

Type: Number or String

A string or number determining how long the animation will run.

**complete**

Type: Function()

A function to call once the animation is complete, called once per matched element.

*(註：The .show([duration][,complete]) method animates the width, height, and opacity of the matched elements simultaneously.)*

#### .hide( ) Returns: jQuery

* + Description: Hide the matched elements.

.hide()

This signature does not accept any arguments.

hide( [duration ] [, complete ] )

**duration (default: 400)**

Type: Number or String

A string or number determining how long the animation will run.

**complete**

Type: Function()

A function to call once the animation is complete, called once per matched element.

#### .fadeIn( [duration ] [, complete ] ) Returns: jQuery

* + Description: Display the matched elements by fading them to opaque.

.fadeIn( [duration ] [, complete ] )

**duration (default: 400)**

Type: Number or String

A string or number determining how long the animation will run.

**complete**

Type: Function()

A function to call once the animation is complete, called once per matched element.

#### .slideDown( [duration ] [, complete ] ) Returns: jQuery

* + Description: Display the matched elements with a sliding motion.
  + 用法同上

#### .fadeOut( [duration ] [, complete ] ) Returns: jQuery

* + Description: Hide the matched elements by fading them to transparent.

.fadeOut( [duration ] [, complete ] )

**duration (default: 400)**

Type: Number or String

A string or number determining how long the animation will run.

**complete**

Type: Function()

A function to call once the animation is complete, called once per matched element.

#### .slideUp( [duration ] [, complete ] ) Returns: jQuery

* + Description: Hide the matched elements with a sliding motion.
  + 用法同上

#### .animate( ) Returns: jQuery

* + Description: Perform a custom animation of a set of CSS properties.

.animate( properties [, duration ] [, easing ] [, complete ] )

**properties**

Type: PlainObject

An object of CSS properties and values that the animation will move toward.

**duration (default: 400)**

Type: Number or String

A string or number determining how long the animation will run.

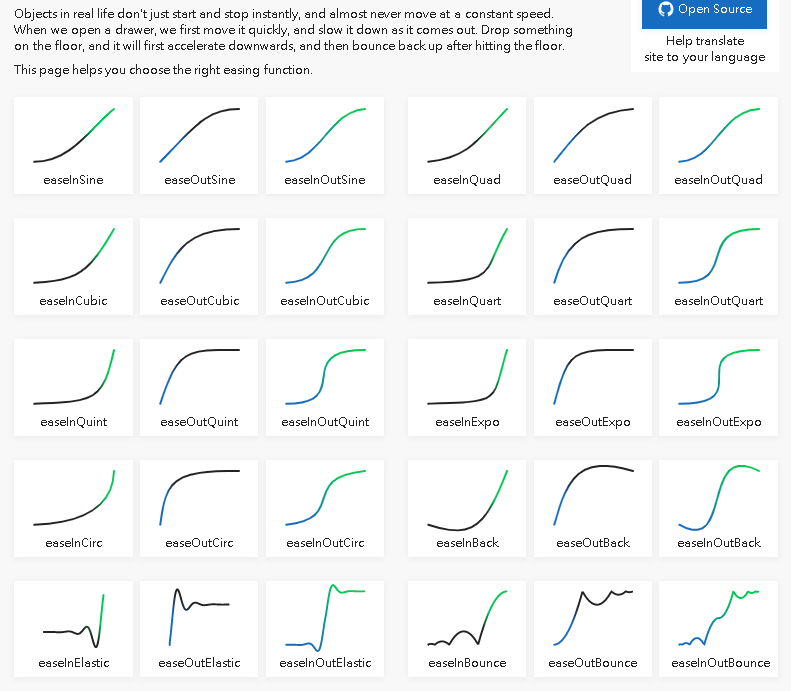
**easing (default: swing)**

Type: String

A string indicating which easing function to use for the transition.

|  |  |  |
| --- | --- | --- |
| 1. linear  2. swing  3. easeInQuad  4. easeOutQuad  5. easeInOutQuad  6. easeInCubic  7. easeOutCubic  8. easeInOutCubic  9. easeInQuart  10. easeOutQuart  11. easeInOutQuart | 12. easeInQuint  13. easeOutQuint  14. easeInOutQuint  15. easeInExpo  16. easeOutExpo  17. easeInOutExpo  18. easeInSine  19. easeOutSine  20. easeInOutSine  21. easeInCirc  22. easeOutCirc | 23. easeInOutCirc  24. easeInElastic  25. easeOutElastic  26. easeInOutElastic  27. easeInBack  28. easeOutBack  29. easeInOutBack  30. easeInBounce  31. easeOutBounce  32. easeInOutBounce |

<https://easings.net/>

****

**complete**

Type: Function()

A function to call once the animation is complete, called once per matched element.

#### 範例：jq\_T04.html

jq\_T04.html

<!DOCTYPE html>

<html>

<head>

<title>特效與動畫</title>

<style type=*"text/css"*>

*#container* {margin:*50px*;}

**h2** {

background: *url(images/open.png) no-repeat 0 11px*;

padding: *10px 0 0 25px*;

cursor: *pointer*;

}

**h2***.close* {

background-image: *url(images/close.png)*;

}

*.answer* {

margin-left: *25px*;

}

*#dashboard* {

width: *70px*;

background-color: *#444*;

padding: *20px 20px 0 20px*;

position: *absolute*;

left: *-92px*;

z-index: *100*;

}

*#dashboard* **img** {

margin-bottom: *20px*;

border: *1px solid rgb(0, 0, 0)*;

}

</style>

<link rel=*"stylesheet"* type=*"text/css"*

href=*"plugins/jQuery UI 1.12.1/stylesheets/base/jquery-ui.css"* />

<script type=*"text/javascript"* src=*"javascripts/jquery-3.4.1.min.js"*></script>

<script type=*"text/javascript"* src=*"plugins/jQuery UI 1.12.1/jquery-ui.min.js"*></script>

<script type=*"text/javascript"*>

//Dashboard(儀錶板)

$(**function**() {

$('#dashboard').hover(**function**() {

$(**this**).stop().animate({

left : '0',

backgroundColor : 'rgba(51,51,51,0.3)'

}, 500, 'easeInSine'); // end animate

}, **function**() {

$(**this**).stop().animate({

left : '-92px',

backgroundColor : '#444'

}, 1500, 'easeOutBounce'); // end animate

}); // end hover

});

//FAQ

$(**function**() {

$('.answer').hide();

$('h2').click(**function**() {

**if** ($(**this**).hasClass('close')) {

$(**this**).next().fadeOut();

$(**this**).removeClass('close');

} **else** {

$(**this**).next().slideDown();

$(**this**).addClass('close');

}

});

})

//註冊: 點擊dashboard的<img>->則切換顯示/隱藏對應順序的解答

$(**function**(){

$("#dashboard img").each(**function**(i,e){

$(**this**).click(**function**(){

$("h2:eq("+i+")").click();

});

});

})

</script>

</head>

<body>

<div id=*"container"*>

<div id=*"dashboard"*>

<img src=*"images/small/blue\_h.jpg"* width=*"70"* height=*"70"* alt=*"blue"*>

<img src=*"images/small/green\_h.jpg"* width=*"70"* height=*"70"* alt=*"green"*>

<img src=*"images/small/orange\_h.jpg"* width=*"70"* height=*"70"*

alt=*"orange"*>

<img src=*"images/small/purple\_h.jpg"* width=*"70"* height=*"70"* alt=*"purple"*>

<img src=*"images/small/red\_h.jpg"* width=*"70"* height=*"70"* alt=*"red"*>

</div>

<h1>A One Page FAQ</h1>

<h2>I've heard that JavaScript is the long-lost fountain of

youth. Is this true?</h2>

<div class=*"answer"*>

<p>Why, yes it is! Studies prove that learning JavaScript

freshens the mind and extends life span by several hundred years.

(Note: some scientists disagree with these claims.)</p>

</div>

<h2>Can JavaScript really solve all of my problems?</h2>

<div class=*"answer"*>

<p>Why, yes it can! It's the most versatile programming language

ever created and is trained to provide financial management advice,

life-saving CPR, and even to take care of household pets.</p>

</div>

<h2>

Is there nothing JavaScript <em>can&#8217;t</em> do?

</h2>

<div class=*"answer"*>

<p>Why, no there isn&#8217;t! It&#8217;s even able to write its

own public relations-oriented Frequently Asked Questions pages. Now

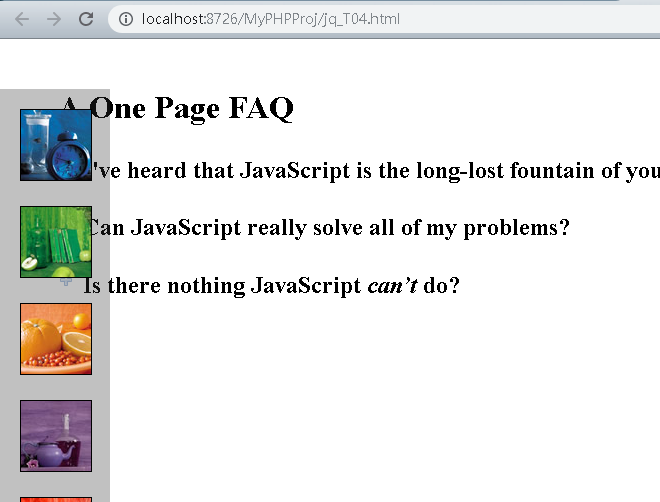
that&#8217;s one smart programming language!</p>

</div>

</div>

</body>

</html>



### 使用AJAX與網站伺服器互動

#### AJAX簡介

* + AJAX這個名詞最早是在2005年被創造出來，當初創造者是希望藉由這個名詞勾勒出Google所推出的一系列新網站(如Google Map、Gmail等)。
  + AJAX全稱為「Asynchronous JavaScript and XML」(非同步JavaScript和XML)，是一種建立互動式網頁應用的網頁開發技術。
  + 使用Ajax的最大優點，就是能在不更新整個頁面的前提下維護資料。這使得Web應用程式更為迅捷地回應使用者動作，並避免了在網路上發送那些沒有改變過的資訊。
  + 傳統的Web應用程式允許用戶端填寫表單，當送出表單時就向Web伺服器發送一個請求。伺服器接收並處理傳來的表單，然後送回一個新的網頁，這個做法浪費了許多頻寬，因為在前後兩個頁面中的大部分HTML碼往往是相同的。
  + AJAX應用可以僅向伺服器發送並取回必需的資料，並在用戶端採用JavaScript處理來自伺服器的回應。
  + 因為在伺服器和瀏覽器之間交換的資料大量減少，所以我們就能看到更快的伺服器回應的結果。
  + 同時很多的處理工作可以在發出請求的用戶端機器上完成，所以Web伺服器的處理時間也減少了。

#### jQuery.ajax( [settings ] ) Returns: jqXHR

* + Description: Perform an asynchronous HTTP (Ajax) request.

jQuery.ajax( [settings ] )

**settings**

Type: PlainObject

A set of key/value pairs that configure the Ajax request.

All settings are optional.

type (default: 'GET')

Type: String

An alias for method. You should use type if you're using versions of jQuery prior to 1.9.0.

url (default: The current page)

Type: String

A string containing the URL to which the request is sent.

data

Type: PlainObject or String

Data to be sent to the server.

dataType (default: Intelligent Guess (xml, json, script, or html))

Type: String

The type of data that you're expecting back from the server.

success

Type: Function( Anything data )

A function to be called if the request succeeds.

#### 範例：jq\_T05.html(page I)

jq\_T05.html(第一頁)

<!DOCTYPE html>

<html>

<head>

<title>使用AJAX與網站伺服器互動</title>

<style>

**h1** {

font-size: *32px*;

color: *black*;

text-shadow: *1px 1px 2px rgba(0,0,0,.75)*;

border-bottom: *solid 1px rgba(0,0,0,.5)*;

margin-bottom: *0.75em*;

}

*#p1* **form** {

padding: *10px 10px 10px 10px*;

}

*#p1* **label** {

display: *inline-block*;

width: *100px*;

text-align:*right*;

margin: *0 15px 0 0*;

color: *black*;

}

*#p1* **input** {

font-size: *14px*;

}

*#p1* *#button* {

margin-left: *115px*;

}

*#p1* *#fail* {

color: *red*;

}

</style>

<link rel=*"stylesheet"* type=*"text/css"*

href=*"plugins/jQuery UI 1.12.1/stylesheets/base/jquery-ui.css"* />

<script type=*"text/javascript"*

src=*"javascripts/jquery-3.4.1.min.js"*></script>

<script type=*"text/javascript"*

src=*"plugins/jQuery UI 1.12.1/jquery-ui.min.js"*></script>

<script type=*"text/javascript"*>

$(**function** () { $("#tabs").tabs(); });

//----------------p1-----------------------//

$(**function** () {

$('#p1 #login').submit(**function** () {

**var** formData = $(**this**).serialize(); //The .serialize() method creates a text string

in standard URL-encoded notation. 例如: 'username=kitty&password=123'

$.ajax( {

type: 'post', //An alias for method. (jQuery 1.9.0以前版本須使用type)

url: 'AjaxServer.php', //A string containing the URL to which the request is sent.

data: formData, //Data to be sent to the server.

//contentType:'application/x-www-form-urlencoded; charset=UTF-8',

success: //A function to be called if the request succeeds.

**function**(data){

**if** (data == 'pass') {

$('#p1 #formwrapper').html('<p>您已成功登入了...</p>');

} **else** {

**if** (! $('#fail').length) {

$('#p1 #formwrapper p:last').append(

'<span id="fail">登入資訊不正確. 請再試一次...</span>');

}

}

}

}

);

**return** **false**;

}); // end submit

});

(略)

</script>

</head>

<body>

<div id=*"tabs"*>

<ul>

<li><a href=*"#p1"*>Ajax: ajax()</a></li>

<li><a href=*"#p2"*>Ajax: getJSON()</a></li>

</ul>

<div id=*"p1"*>

<h1>Login</h1>

<div id=*"formwrapper"*>

<form method=*"get"* action=*"AjaxServer.php"* id=*"login"*>

<p>

<label for=*"username"*>使用者名稱:</label>

<input type=*"text"* name=*"username"* id=*"username"*>

</p>

<p>

<label for=*"password"*>密碼: </label>

<input type=*"text"* name=*"password"* id=*"password"*>

</p>

<p>

<input type=*"submit"* id=*"button"*>

</p>

</form>

</div>

</div>

(略)

</div>

</body>

</html>

AjaxServer.php

<?php

**if**(**$\_POST**['username']=='hello' && **$\_POST**['password']=='kitty')

**echo** 'pass';

**else**

**echo** 'fail';

?>



#### 以AJAX向伺服器端請求JSON資料

* JSON導論
* JSON(JavaScript Object Notation)是一種由道格拉斯·克羅克福特(Douglas Crockford)構想設計、輕量級的資料交換語言，以文字為基礎，且易於讓人閱讀。儘管JSON是Javascript程式語言的一個子集，但JSON是獨立於語言的文字格式，並且採用了類似於C語言家族的一些習慣。
* 目前很多程式語言都支援JSON格式資料的生成和解析。JSON的官方MIME類型是application/json，副檔名是.json。
* JSON 的基本資料型別有：

1. Number

2. String

3. Boolean

4. Array

5. Object

6. null

* JSON主要建構在兩個資料結構上：

1. Object

2. Array

* JSON資料範例：

{

"firstName": "John",

"lastName": "Smith",

"isAlive": **true**,

"age": 25,

"height\_cm": 197.6,

"address": {

"streetAddress": "21 2nd Street",

"city": "New York",

"state": "NY",

"postalCode": "10021-3100"

},

"phoneNumbers": [

{

"type": "home",

"number": "212 555-1234"

},

{

"type": "office",

"number": "646 555-4567"

}

],

"children": [],

"spouse": **null**

}

#### jQuery.getJSON( url [, data ] [, success ] ) Returns: jqXHR

* + Description: Load JSON-encoded data from the server using a GET HTTP request.

jQuery.getJSON( url [, data ] [, success ] )

**url**

Type: String

A string containing the URL to which the request is sent.

**data**

Type: PlainObject or String

A plain object or string that is sent to the server with the request.

**success**

Type: Function( PlainObject data )

A callback function that is executed if the request succeeds.

#### 範例：jq\_T05.html(page II)

jq\_T05.html (第二頁)

<!DOCTYPE html>

<html>

<head>

<title>使用AJAX與網站伺服器互動</title>

(略)

<script type=*"text/javascript"*>

$(**function** () { $("#tabs").tabs(); });

(略)

//----------------p2-----------------------//

$(**function** () {

$.getJSON("data/Emp.json", **function**(data){

$.each(data, **function** (index, value) { //index:Emp陣列之索引 value:Emp物件

$("#p2 ul").append("<li>" + index + " : "

+ value.EmpId + " " + value.Name + " " + value.Age

+ "歲" + "</li>");

});

});

//上式相當於下式

$.ajax({

type: "GET",

url: "data/Emp.json",

dataType: "json", //The type of data that you're expecting back from the

server. (default: Intelligent Guess (xml, json, script, or html))

success: **function** (data) {

$.each(data, **function** (index, value) {

$("#p2 ul").append("<li>" + index + " : " + value.EmpId + " "

+ value.Name + " " + value.Age + "歲" + "</li>");

});

}

});

});

</script>

</head>

<body>

<div id=*"tabs"*>

<ul>

<li><a href=*"#p1"*>Ajax: ajax()</a></li>

<li><a href=*"#p2"*>Ajax: getJSON()</a></li>

</ul>

(略)

<div id=*"p2"*>

<h1>Employees載自JSON檔</h1>

<ul></ul>

</div>

</div>

</body>

</html>

Emp.json

[

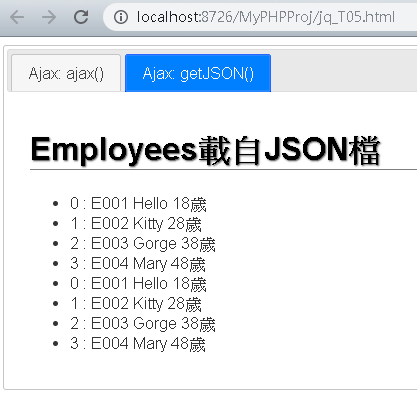
{"EmpId":*"E001"*,"Name":*"Hello"*,"Age": 18},

{"EmpId":*"E002"*,"Name":*"Kitty"*,"Age": 28},

{"EmpId":*"E003"*,"Name":*"Gorge"*,"Age": 38},

{"EmpId":*"E004"*,"Name":*"Mary"*,"Age": 48}

]



## jQuery UI常用元件

### DatePicker(日期選擇器)

$(" input[type=text]")**.datepicker( );**

* 運作模式選項
  + - dateFormat: 設定日期格式
      * 日期格式設定參數參考下面說明
      * 常用設定實例: 'yy/mm/dd'
    - defaultDate: 設定預設日期
      * 預設: 當日
    - showOn: 設定顯示之時機
      * 設定值: "focus"(預設)、"button"、"both"
    - buttonImage: 設定按鈕之圖示
      * 例如: 'images/calendar.jpg'
    - buttonText: 設定按鈕之文字
      * 若buttonImage同時設定，則本選項變成tooltip
    - changeYear: 設定年度是否可使用下拉選單點選
      * 預設: false
    - minDate: 設定可點選之最小日期
      * 預設null (即不限)
    - maxDate: 設定可點選之最大日期
      * 預設null(即不限)
* 日期格式(dateFormat)設定參數

d - day of month (no leading zero)

dd - day of month (two digit)

o - day of the year (no leading zeros)

oo - day of the year (three digit)

D - day name short

DD - day name long

m - month of year (no leading zero)

mm - month of year (two digit)

M - month name short

MM - month name long

y - year (two digit)

yy - year (four digit)

* 日期選擇器中文化

載入 jquery.ui.datepicker-zh-TW.js



### Dialog(對話框)

$("div").dialog( );

* 運作模式選項
  + - width: 設定對話框之寬度
      * 單位: pixel
      * 預設: 300
    - modal: 設定對話框是否具modal模式
      * 預設: false
    - autoOpen: 設定對話框是否自動開啟
      * 預設: true
    - title: 設定對話框標題列之標題
      * 預設: null
    - buttons: 設定對話框欲顯示之按鈕
      * 預設: []
      * 常用設定如:

[ { /\* #1按鈕 \*/

**text**: '確定',

**icons**: { primary: "ui-icon-check" }, /\*設定按鈕圖示\*/

/\*primary:對應位於按鈕左側之圖示\*/

**click**: function () { } /\*註冊click事件處理器\*/

},

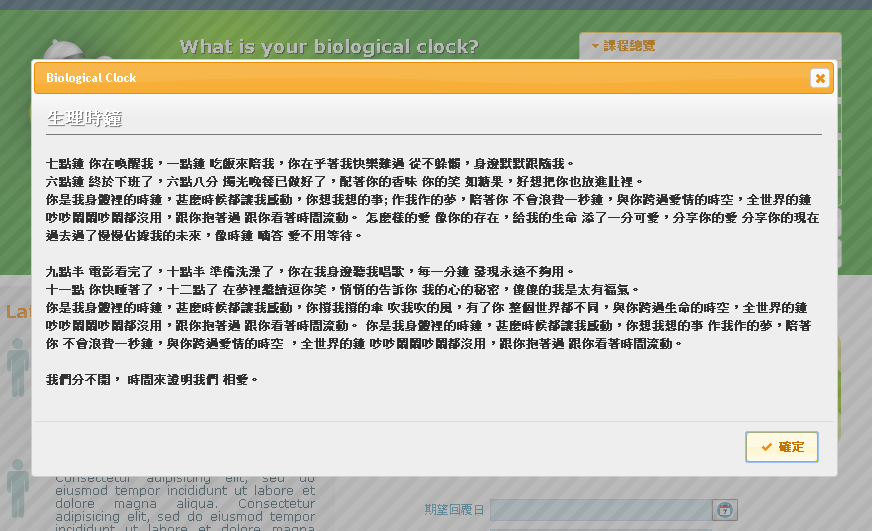
{ /\* #2按鈕 \*/

略

}

]





### Accordion(手風琴選單)

$("div").accordion();

* 須搭配下列HTML元素結構使用

<div>

<div>群組標題1</div>

<ul>

<li><a href="#">option1</a></li>

<li><a href="#">option2</a></li>

</ul>

<div>群組標題2</div>

<ul>

<li><a href="#">option1</a></li>

<li><a href="#">option2</a></li>

</ul>

</div>



### Draggable 與Droppable

$("\*").draggable();

* 運作模式選項
  + - helper: 設定元素被拖曳的模式
      * 設定值: "original"(預設) | "clone"
    - opacity: 設定被拖曳元素的不透明度
      * 設定值: 0~1

$("\*").droppable();

* 運作模式選項
  + - drop: 設定drop事件處理器
      * 設定值: function(event,ui){ }



### ToolTip(工具提示框)

$(document).tooltip();

* 運作模式選項
  + - items: 設定網頁中欲提供工具提示框功能之元素
      * 設定值: 選擇器字串
      * 預設值: "[title] "
    - content: 設定工具示框顯示的內容
      * 設定值: function或string
      * 預設值: function (回傳元素之title屬性值)
    - show|hide: 設定工具提示框顯示|隱藏之特效
      * 預設值:

{ **effect:** 'fadeIn|fadeOut', **duration:** 'normal',

**delay:** 0, **easing:** 'swing' }

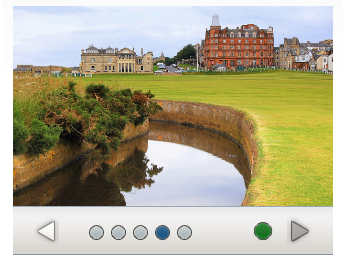
* + - * 其中effect可設定之值:
* jQuery: show|hide、fadeIn|fadeOut、slideDown|slideUp
* jQuery UI: Blind、Bounce、Clip、Drop、Explode、Fade、Fold、Highlight、Puff、Pulsate、Scale、Shake、Size、Slide、Transfer等 (參考<https://jqueryui.com/effect/>)

****

### AnythingSlider(內容輪播器/Content Slider)外掛程式

$("div").anythingSlider();

* 運作模式選項
  + - theme: 設定佈景主題
      * default(預設) (需載入 "anythingslider.css" 樣式表)
      * cs-portfolio (需載入 "theme-cs-portfolio.css" 樣式表)



* + - autoPlay：設定是否自動撥放
      * 預設: false
    - delay: 設定輪播之停頓時間
      * 單位:毫秒
      * 預設: 3000
    - buildArrows: 設定是否顯示「上一張」與「下一張」箭頭
      * 預設: true
    - buildNavigation:設定是否顯示每個面板之連結鈕
      * 預設: true
    - buildStartStop:設定是否顯示「Start」與「Stop」按鈕
      * 預設: true
    - vertical: 設定輪播方向是否「由左往右」改為「由上往下」
      * 預設: false

### 範例：jq\_T06.html

<!DOCTYPE html>

<html>

<head>

<title>jQueryUI常用元件</title>

<style type=*"text/css"*>

**h1** {

font-size: *32px*;

color: *black*;

text-shadow: *1px 1px 1px rgba(0,0,0,.5)*;

border-bottom: *solid 1px rgba(0,0,0,.5)*;

margin-bottom: *0.75em*;

}

*#right\_content* {

font-family: *Arial*;

}

*#right\_nav* **ul** {

overflow: *hidden*;

}

*.tel* {

font-size: *1.5em*;

cursor: *default*;

}

</style>

<link rel=*"stylesheet"* type=*"text/css"* href=*"stylesheets/greefies/style.css"* />

<link rel=*"stylesheet"* type=*"text/css"*

href=*"plugins/jQuery UI 1.12.1/stylesheets/ui-lightness/jquery-ui.css"* />

<link rel=*"stylesheet"* href=*"plugins/anythingSlider1.9.2/css/theme-cs-portfolio.css"*>

<style type=*"text/css"*>

/\* for datepicker > buttonImage \*/

**button***.ui-datepicker-trigger* { padding:*1px 3px*; height:*22px*; }

/\* for anythingSlider \*/

*#slider* {

width: *320px*;

height: *200px*;

}

</style>

<script type=*"text/javascript"* src=*"javascripts/jquery-3.4.1.min.js"*></script>

<script type=*"text/javascript"* src=*"plugins/jQuery UI 1.12.1/jquery-ui.min.js"*></script>

<script type=*"text/javascript"* src=*"javascripts/jquery.ui.datepicker-zh-TW.js"*></script>

<script type=*"text/javascript"*

src=*"plugins/anythingSlider1.9.2/js/jquery.anythingslider.min.js"*></script>

<script type=*"text/javascript"*>

$(**function** () {

//日期選擇器

$("input[name=date]").datepicker({

dateFormat: 'yy/mm/dd', showOn: "both",

buttonImage: 'images/calendar.png', buttonText:'日曆'

});

//對話框

**var** dialog1 = $("<div></div>")

.load('data/BiologicalClock.html #div1')

.dialog({

width: 800, modal: **true**, autoOpen: **false**, title: 'Biological Clock',

buttons: [{ text: '確定', icons:{'primary':'ui-icon-check'},

click: **function** () { $(**this**).dialog('close') } }]

});

$(".green\_box .read\_more a").click(**function** () {

$(**this**).blur();

dialog1.dialog('open');

**return** **false**;

});

//手風琴選單

$("#right\_nav").accordion();

//Draggable與Droppable

$(".tel").draggable({ helper: 'clone', opacity: 0.5 });

$("#contact\_form input[name='contacted\_phone']").droppable({

drop: **function** (event, ui) {

$(**this**).val(ui.draggable.text());

}

});

//工具提示框

$(document).tooltip(

{

items: "[title],img", //預設[title]

content: **function** () {

**var** element = $(**this**);

**if** (element.is("[title]"))

**return** element.attr('title');

**else** **if** (element.is("img")) {

**return** element.attr('alt');

}

},

hide: { effect: 'explode', delay: 250 }

}

);

//輪播器

$("#slider").anythingSlider({ theme: 'cs-portfolio', autoPlay: **true**,

buildArrows: **false**, buildNavigation: **false**, buildStartStop: **false** });

});

</script>

</head>

<body>

<!--Greefies Css Template-->

<div id=*"main\_container"*>

<div id=*"header"*>

<div id=*"logo"*>

<img src=*"stylesheets/greefies/images/logo.gif"*

alt=*"greefies solution"* border=*"0"* />

</div>

<div id=*"menu"*>

<ul>

<li><a href=*"home.html"* title=*""*>home</a></li>

(略)

</ul>

</div>

</div>

<div class=*"green\_box"*>

<div class=*"clock"*>

<img src=*"stylesheets/greefies/images/clock.png"* alt=*"生理時鐘"* />

</div>

<div class=*"text\_content"*>

<h1>What is your biological clock?</h1>

<p class=*"green"*>

"Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do

(略)."

</p>

<div class=*"read\_more"*>

<a href=*"#"* title=*"查看詳細內容"*>read more</a></div>

</div>

<div id=*"right\_nav"*>

<div>課程總覽</div>

<ul>

<li><a href=*"#"*>程式開發</a></li>

<li><a href=*"#"*>資訊安全</a></li>

<li><a href=*"#"*>網路通訊</a></li>

<li><a href=*"#"*>雲端運算</a></li>

</ul>

<div>近期課程</div>

<ul>

(略)

</ul>

<div>客戶服務</div>

<ul>

(略)

</ul>

</div>

</div>

<!--end of green box-->

<div id=*"main\_content"*>

<div id=*"left\_content"*>

<h2>Latest News</h2>

<div class=*"news"*>

<div class=*"news\_icon"*>

<img src=*"stylesheets/greefies/images/testimonials\_icon.gif"*

alt=*""* title=*""* />

</div>

<div class=*"news\_content"*>

<span>"Lorem ipsum dolor sit amet</span><br />

Consectetur adipisicing elit, sed do eiusmod (略).

</div>

</div>

<div class=*"read\_more\_link"*><a href=*"#"*>read more</a></div>

<div class=*"news"*>

<div class=*"news\_icon"*>

<img src=*"stylesheets/greefies/images/testimonials\_icon.gif"*

alt=*""* title=*""* />

</div>

<div class=*"news\_content"*>

<span>"Lorem ipsum dolor sit amet</span><br />

Consectetur adipisicing elit, sed do eiusmod tempor (略)i.

</div>

</div>

<div class=*"read\_more\_link"*><a href=*"#"*>read more</a></div>

<div class=*"news"*>

<div id=*"slider"*>

<div>

<a href=*"#"*>

<img src=*"images/large/slide1.jpg"*

width=*"320"* height=*"200"* alt=*"slide1"*>

</a>

</div>

<div>

<a href=*"#"*>

<img src=*"images/large/slide2.jpg"*

width=*"320"* height=*"200"* alt=*"slide2"*>

</a>

</div>

(略)

</div>

</div>

</div>

<!--end of left content-->

<div id=*"right\_content"*>

<h2>Contact Information</h2>

<div class=*"contact\_info\_box"*>

<div class=*"contact\_info\_title"*>Dolore magna aliqua</div>

<div class=*"contact\_info"*>

<img src=*"stylesheets/greefies/images/phone\_icon.gif"*

alt=*""* title=*"連絡電話"* class=*"box\_img"* />

<span class=*"tel"*>07-9699885#123</span><br />

<span class=*"tel"*>07-9699885#456</span>

</div>

(略)

<div class=*"adress"* data-geo=*""* style="width: *200px*;">

高雄市中正四路211號8F-1

</div>

</div>

<h2>Contact form</h2>

<div id=*"contact\_form"*>

<div class=*"form\_row"*>

<label>期望回覆日</label>

<input style="display: *inline*" type=*"text"*

name=*"date"* class=*"contact\_input"* />

</div>

<div class=*"form\_row"*>

<label>姓名:</label><input type=*"text"*

name=*"name"* class=*"contact\_input"* />

</div>

(略)

<div class=*"form\_row"*>

<label>曾洽電話:</label><input type=*"text"*

name=*"contacted\_phone"* class=*"contact\_input"* />

</div>

(略)

<div class=*"send"*><a href=*"#"* title=*"寄發聯絡信"*>

send</a></div>

</div>

</div>

<!--end of right content-->

</div>

<!--end of main content-->

(略)

</div>

<!--end of main container-->

</body>

</html>