JR

標題

<h1> 主標題 <h2> 副標題 <h3>小標題

無序清單 、 有序清單

<ul>

<li>清單</li>

<li>清單</li>

<li>清單</li>

</ul>

<ol>

<li>清單</li>

<li>清單</li>

<li>清單</li>

</ol>

內文變化 :

超連結 <a href=" "> </a> / 斜體 <em> </em> 、 <i> </i>

粗體 <strong> </strong>

圖片標籤

圖片 <img src=" " alt=" ">

圖片

<figure>

<img src=" " alt=" ">

<figcaption>(描述文字)圖說</figcaption>

</figure>

選單

 <nav>

<a href="">選單</a>

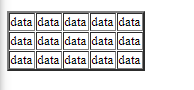
<a href="">選單</a>

<a href="">選單</a>

</nav>

表格

<table border="2"> <!-- border="2" 增加表格邊框線 -->

 <tr>

<td>data</td>

<td>data</td>

<td>data</td>

<td>data</td>

<td>data</td>

</tr>

<tr>

<td>data</td>

<td>data</td>

<td>data</td>

<td>data</td>

<td>data</td>

</tr>

<tr>

<td>data</td>

<td>data</td>

<td>data</td>

<td>data</td>

<td>data</td>

</tr>

</table>

**選取器{**

**屬性: 屬性的值;**

**屬性: 屬性的值;**

**}**

段落文字樣式

P {

color: red; 字體顏色

font-size: 20px; 字體大小

background-color: blue; 字體背景顏色

}

CSS reseat設定 :

\*{

margin : 0;

padding : 0;

}

導覽列呈現

nav{

text-align: center; 選單置中

}

nav a {

color: white;

text-decoration: none; 字體裝飾 : 無

display: inline-block; 字體有間距出來

padding: 6px 20px;

font-size: 16px;

transition: 0.5s; (呈接下方havor摸到後延遲0.5秒)

}

nav a:havor{ (havor:如果nav被碰到會有效果)

background-color: orange; 碰到後背景變橘色

}

一張含有 螢幕擷取畫面 的圖片

描述是以非常高的可信度產生

!important 意義

<https://n.sfs.tw/content/index/10632>

網頁的前端工程師非常頭痛的事，就是同時要處理 IE和 FF等不同的瀏覽器，更糟的是 IE 又分成了 6  7 8 9 10 11 12 ....不同版本，同樣的 CSS在不同的browser或是同browser不同版本中，竟會差異天攘之別，這才是煩人。

好了，來說說 css 中 !important 的意義。

有人是這樣寫的：

看得懂的browser  會處理 !important 的css，不看懂的browser  會忽略。IE一直都不支持這個語法，而其他的瀏覽器都支持。我們就可以利用這一點來分別給IE和其它瀏覽器不同的樣式定義。**(x)**

!important 前面是給Firefox看的**(x)**

其實上面的錯在於沒有把!important 意義搞清楚，先來看以下範例：

1. 都沒使用 !important，在這種情況下，第二個p會覆寫第一個p，所以是紅色。

#example p { color: blue; }  
#example p { color: red; }

2. 一個使用 !important，**因為第一個有!important，所以有更高的優先權，所以會是藍色**。

#example p { color: blue !important; }

#example p { color: red; }

3.  二個都使用 !important，**同1，第二個p覆寫了第一個p，所以是紅色**。

#example p { color: blue !important; }

#example p { color: red !important; }

4.  利用 !important 覆寫 style**，如下範例，所以是藍色**。

CSS:  
#example p { color: blue !important; }

HTML:  
<div id="example"> <p style="color:green;">This paragraph has an inline style</p> </div>

5.  利用 !important 強制鎖定型別**，如下範例，所以是綠色**。

CSS:  
#example p { color: blue !important; }

HTML:  
<div id="example"> <p style="color:green !important;">This paragraph has an inline style</p> </div>

6. 以上基本上出現的結果會和我說的一樣，除了IE6 & 7以外，各版本的IE表現：

Internet Explorer 6 會採用最後一個宣告的屬性，根本不理會 !important，所以範例2 只有IE6,7會是紅色 ，所以我原本都誤會 !important 是程式語言的 NOT important 不重要的意思，事實上反而是相反的意思：

!important 是「更重要」的意思，對新的browser來說，是最高優先的意思；對不支援!important的browser 而言，會忽略無作用。

IE7 必需要宣告 doctype 才會正常顯示。

IE8 就比較正常了。

1. !important 是跟著每個屬性設定，所以要在屬性中：

   color: #123456 !important;   <== ○ 正確寫法指定覆寫  
   color: #123456**;**!important;   <== x 錯誤寫法，覆寫不會生效

2. 類別中有多個屬性，每個屬性都要覆寫，就每個屬性值都要有 !important

 .classname { color:#123456; background:#987654 !important; }     <==  指定覆寫background，color 不受影響  
 .classname { color:#123456 !important; background:#987654 !important; }     <==  指定覆寫background、color

3. 除非你很清楚知道你在覆寫什麼東西，不然少用!important，因為他有可能導致你查錯困難。

4. CSS的權重：

有!important的 element style(設定在tag中的style) >  有!important的css >  element style(設定在tag中的style) > css後寫的 > css先寫的

CSS 中常見的背景 (background) 屬性

<https://www.1keydata.com/css-tutorial/tw/background.php>

一張含有 螢幕擷取畫面 的圖片

描述是以非常高的可信度產生

一張含有 螢幕擷取畫面 的圖片

描述是以非常高的可信度產生

一張含有 螢幕擷取畫面 的圖片

描述是以非常高的可信度產生

一張含有 螢幕擷取畫面 的圖片

描述是以非常高的可信度產生

使用CSS3及HTML製作圖片輪播

<http://blog.shihshih.com/pure-css-slider/>

url、href、src 詳解

<https://segmentfault.com/a/1190000002877022>

半透明值 (alpha 色版) 的 CSS 3 顏色單位

<http://boohover.pixnet.net/blog/post/20556952-%E5%8D%8A%E9%80%8F%E6%98%8E%E8%88%87%E9%80%8F%E6%98%8E%E6%95%88%E6%9E%9C%E8%A8%AD%E8%A8%88-css-rgba%2C-hsla-%E8%89%B2%E5%BD%A9%E5%96%AE%E4%BD%8D-%28>

常用標記標籤

一張含有 文字 的圖片

描述是以非常高的可信度產生

字距 :

letter-spacing : 2 px ;

行高 :

line – height : 50px ;

字體粗細 :

font-weight : 100 ; (100-900 900最粗)

分隔線 :

</hr>

透明色塊 :

rgbs(0 ,128 ,0 ,0.5 ) - 最後的0.5代表透明值50%

透明圖片 :

opacity: 0.5 50%透明度

製作 圖形 或 小面積卷軸 :

overflow : hidden ;

overflow : scroll ;

盒模型 :

box-sizing : border-box; (將寬高指定給border)

box-sizing : content-box (將寬高指定給content)

<input> Type :

資料來源 <http://crazy.molerat.net/learner/cpuroom/programmer/reading.php?filename=http://crazy.molerat.net/learner/cpuroom/programmer/1010407112825.dov>

text (文字輸入)

password (密碼輸入)

hidden (隱藏變數輸入)

radio (單一選擇項目輸入)

checkbox (多重選擇鈕(核取方塊))

reset (表單資料重新輸入鈕)

submit (傳送表單按鈕)

image (圖形送出鈕)

text：設定文字輸入欄位。 可以把任何字元或數字，

甚至於一大堆亂七八糟的符號，

都輸入到該欄位裡頭。

之前再 第一張表單「幻境入門」， 已經學過，

我們可以在那張表單一個名叫 friend 的欄位輸入文字內容，

各位都一定還記得。

再請仔細閱讀這個敘述：

<input type="text" name="user" value="請輸入暱稱" size="35" maxlength="35">

沒錯，真是厲害！

你竟然瞧出來了：

從 「input」 這個標籤名稱你就知道

「再這裡可以輸入資料」，

而且從「type="text"」 你也可以看懂

「這是一個文字輸入欄」，

而它的欄位名稱叫做 "user"，

因為這裡有個屬性寫法是「name="user"」對嗎？

喝呵！所以，

後面的 size 和 maxlength 兩個屬性就不必解釋了吧？

許多留言板或者網頁登入表單第一個文字輸入框經常都像這樣寫的。

而如果你想用 php 知道網頁用戶在文字輸入欄輸入的是什麼內容，

只要你有辦法用 php 抓到該欄位的內容值就行了。

其實這也不會很困難，

以這個實例來說，

我們可以從 「name="user"」得知該欄位的名稱是 "user"，

因此 php 只要呼叫辨書名稱為 $user 就可以知道用戶在這裡面到底寫了什麼？

而只要能知道該變數的內容是什麼，

就也可以透過 php 程序加以楚哩，

或是坐伊些條件判斷。

等我們把基礎的表單設計概念弄清楚之後，

學習 php 處理表單的方式時就會更容易了解。

password：設定密碼輸入欄。 使用某些網頁服務，

你一定常需要在某個登入頁面上輸入密碼吧？

然後你也一定會發現再這樣的輸入欄敲進去的每個字元都變成星號符，

那個欄位用的就是這種輸入類型，

「還用說嗎？

既然是要輸入密碼，

當然就不可以給人家看到啊！」

所以登入頁面的密碼欄位就可以這樣子寫：

<input type="password" name="pwd" value="guest" size="35" maxlength="35">

與上面文字輸入欄的敘述依樣，

這裡的 name= 跟 value= 之後的內容定義也可以任由你自己來決定，

使用 php 處理表單資料，

想知道某位使用者所輸入的密碼是什麼，

也可以用 「echo」這個指令來抓出該欄位名稱的內涵值，

所使用的變數名稱當然也是該欄位 name 屬性所定義的名稱，

在此欄位的 name 是 "pwd"，

所以用 php 「echo $pwd」這個指令，

就可以印出使用者在這個密碼欄位輸入的內容是什麼了。

而如果要判斷某個網頁使用者是否輸入正確的密碼，

只要在 php 程序寫一個條件判斷敘述：

「如果 $pwd 的內容不符合條件就算是登入失敗」。

至於你想設定該文字框框的大小是多少，

也得你自己依照實際狀況而設計囉！，

講究頁面美觀的現代人們，

框框太大跟太小都對視覺美感有影響的。

hidden： 指定該欄位的類型為一個隱藏的文字輸入欄，

這種類型的欄位並不會顯示在表單的頁面上，

網頁使用者也無法直接在頁面上修改該欄的設定值，

因為該屬性主要的作用，

是把網頁伺服器上程式所需要的資料，

透過表單傳送給它能夠應用而已，

好比說一隻小狗需要吃雞肉才有力氣工作，

我們就可以利用 input 的 hidden 屬性，

設定隱藏資料輸入欄位，

把雞肉從這個欄位傳過去給那隻小狗吃。

如果你曾經檢視過學習的故鄉原始碼，

該不難發現它的發表文章或者留言板表單也有許多 hidden 欄位，

那些東西其實都在告訴伺服器 php 的處理程序該如何運作罷了，

例如：

<input type="hidden" name="date" value="民國101年4月4日（星期三）14點36分">

這是一個名為 "date" 的隱藏輸入欄，

從 value= 我們可以發現，

date 的設定值為 "民國101年4月4日（星期三）14點36分"，

因此如果表單送出給某個 php 程序的時候，

使用 「echo $date」這個指令，

瀏覽器畫面上就會呈現「民國101年4月4日（星期三）14點36分」這個內容，

日後我們學到比較複雜的程序運作時就會知道了。

表單還有另外兩種非常重要的資料輸入按鈕設定， 即選項按鈕與核取方塊兩個類型，

另外還有下拉式方塊標籤設置，

文字編輯區塊的定義方式，

這些部分需要講解的地方很多，

我想個別用專門的一篇文章與大家分享可好？

此外，

還剩下兩個 type= 的類型，

本文結束前先在這裡告訴大家，

一個是清除表單資料鈕，

另一個則是表單送出鈕。

當你把表單資料填寫完畢， 發現想要全部重新填寫的時候，

假如要填寫的東西很多，

每個欄位都要清除才能再重新填入多麻煩呀！

如果網頁設計者設計表單的時候有設置表單清除資料鈕：

<input type="reset" name="reset" value="清除重寫" >

如此你只要在該按鈕點一下，

剛才填寫的資料就會消失了，

此時你也就可以再重新填寫表單。

話雖如此，

但是我覺得這種按鈕好像已經很少人使用，

你想想呀！

如果表單填寫完畢真需要再重新填寫，

直接關閉表單頁面或者按 f5 重新載入頁面不就行了，

有時我們辛辛苦苦把表單資料填寫完畢不小心按到 reset，

前功盡棄，

徒勞無功，

一切罔然，

那又何必呢？

所以，

為了部讓網頁使用者因為不小心錯按清除按鈕，

還要重填資料而感到麻煩，

甚至於乾脆放棄不寫算了，

倒不如不要設置此按鈕還比較理想。

聽鴿子說有一陣子許多人辛辛苦苦在故鄉上面把一篇精彩的文章寫好，

結果不小心或者某些原因，

總之反正就是正想把文章完成的時候發現剛才的辛苦都不見了，

非常氣人，

乾脆不寫了，

他懷疑就是 reset 搞的鬼，

因此後來也就沒有在故鄉的任何一個表單設定此按鈕。

表單填寫完畢， 最最要緊的就是要把填寫好的資料送出去，

送到 action 所指定的網頁伺服器檔案處理。

假如沒有設置表單送出按鈕，

資料無法送出的話，

填寫表單就變成毫無意義了。

雖然你也可以在文字輸入框把資料填好之後按一次回車，

用那種方式也可以送出表單，

但那可能會導致難以想像的後果。

傳送表單按鈕的語法是：

<input type="submit" name="submit" value="填寫完畢">

name 跟 value 這兩個屬性應該不必再介紹了， 尤其 value，

那是要呈現給網頁使用者可以在畫面上看到的按鈕名稱，

事實上真正的按鈕名稱應該是 name 的設定值，

例如：

name="submit"

表示這個欄位的名稱就是 "submit"。

而 value 則是該按鈕的內涵值，

例如：

value="填寫完畢"

當網頁使用者點一下頁面上所呈現的「填寫完畢」，

點此按鈕表單就被送回 action 所指定的伺服器程式檔處理，

在 php 只要知道表單的每個欄位名稱，

就能使用該欄位名稱所產生的變數名稱處理表單資料了，

例如我們可以在 php 寫一段條件敘述，

判別表單是否已經傳送到伺服器：

<?php

if($submit == "填寫完畢")

{

　echo "<title>";

　echo "嗨！";

　// 下依行所呈現的就是網頁使用者再

<input type="text" name="user" value="請輸入暱稱" size="35" maxlength="35">

這個欄位所輸入的內容

　echo $user;

// 注意：如果沒有輸入任何內容就傳送表單的話，$user 的內容就是該欄位 value 所定義的內容

　echo "歡迎光臨幻境之家";

　echo "</title>";

　echo "今天是：";

　// 下面的內容就是

<input type="hidden" name="date" value="民國101年4月4日（星期三）14點36分">

這個隱藏欄位資料輸入欄丟過來的內容

　echo $date;

　...; // 更多 php 程序執行代碼我們會在日後篇章用到的時候學會

} // if($submit == "填寫完畢")

?>