

#2019.7.7 開始一更 參考 <https://www.codecademy.com/learn/learn-html>  
#2019.7.10 完成一更，新增 CSS 會用到的 Tag、attribute

前言:

What is html

Html 是網頁呈現的骨幹，基礎上決定網頁如何呈現

\*HTML 代表 HyperTextMarkupLanguage (超文本標記語言)

\*Markup language 是定義如何呈現 raw text 的結構

\*HyperText 表示電腦可取用其他 Text 或 links (像 Hyperlinks 一樣)

目錄

A. [HTML 結構](#)

B. [常用基本 HTML element](#)

C. [Table 元素細解](#)

D. [Form 元素細解](#)

A 標準 HTML 結構

big picture

1. 文件宣告

開頭要打上此行，告訴瀏覽器此文件是 HTML

`<!DOCTYPE html>`

告訴瀏覽器此檔案的類型、HTML 版本(預設 HTML5)

2. 主結構

加入 `<html>`

...

`</html>`

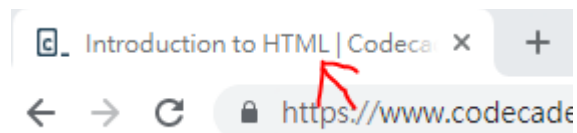
表示 html 主結構

3. 標頭

在 `<html>` 後會有 `<head>...</head>`

以告訴瀏覽器這個 website 的資訊

例如此網頁名(顯示於瀏覽器頁上)



4. 內容

`<body>...</body>`

主要會呈現的內容==>只要在網頁上呈現的都寫在這

5. 內部連結(internal link)

超連結 `<a>...</a>` 等通常都是連到外部網站，而我們通常也會有內部網頁可以連結

通常我們會建資料夾存放 HTML 檔，且通常存於同一層，才能使用相對路徑

EX:

```
<a href=" ./contact.html">Contact</a>
```

## 6.網頁內超連結:

搭配 **id** 屬性與 `<a href="#yourID"> ...</a>`

\*id 要加#就可以用 a 連結!!

\*通常搭配 `<ul>` 存放網頁內超連結

## 7.comment

`<!-- your comment here -->`

## 8.基本結構例子

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <!-- title ...資訊-->
```

```
  </head>
```

```
  <body>
```

```
    <!-- 網頁顯示內容-->
```

```
  </body>
```

```
</html>
```

我做了 HTML 範例檔可參考

## B.HTML element

(a) 簡介:

### 1.元素組成

Html 是由 element 組成, 而 element 由 **opening tag**、**content**、**closing tag** 組成(有特例會沒有 closing tag!, 特例即為不含內文的 element, 例如 `<br>`)

ex:

```
<p> hello world </p>
```

(\*Note:html tag 就是 html element 的名稱)

### 2.元素結構

元素的結構是 Nested 的

在外的是 parent element 在內的是 child element (\*Note:有繼承關係喔)

ex:

```
<body>
  <p>This paragraph is a child of the body</p>
</body>
```

為了易讀性要善用縮排

(b)element 常用種類

===== <body>使用 =====

### 1.<body>

只有在 body element 中的 content 會呈現於網頁主體中

## 2.<h1> ~ <h6>

代表標題(heading)

<h1> 最大，通常用於主標題

大小隨數字遞減

## 3.<div>

代表分割(division)

可以封裝任何元素在其中

簡單來說就是分區塊的

(只對程式分區塊==>增加易讀性)

(在後期會提到更多用處)

(無關排版，只是可以分區加入屬性...等)

## 4.屬性

屬性是加在 **opening tag** 中，所有 element 都可以有屬性

屬性可以使 element 做更多事 (換 style、註解、link ....)，提供額外資訊

屬性由 **Name** 和 **value** 組，格式為 Name = "value"

例如：

```
<div id="intro">
  <h1>Introduction</h1>
</div>
```

常用屬性：

**id**: 幫助我們為同名但不同的元素**做記號區分** (例如:最常使用於 div)

**class**:幫助 CSS 設定，可一次賦予多值

例如<p class="gold green">...</p>

## 5.<p>

paragraph

用來呈現文章內容

\*排版用

<span>..</span>:類似一段文字內的 div，可以讓這段文字獨立一組施加不同的效果

<em>...</em>:斜體強調

<strong>...</strong>:粗體強調

<br> 換行(link breaks)，注意沒有 closing tag

## 6<ul> & <li>

unorder list and list item

語法

效果

```
<ul>
  <li>Limes</li>
  <li>Tortillas</li>
  <li>Chicken</li>
</ul>
```

- Limes
- Tortillas
- Chicken

7.<ol>& <li>  
orderlist (just like ul)

語法

```
<ol>  
<li>Russia</li>  
<li>United States</li>  
<li>Canada</li>  
</ol>
```

效果

1. Russia
2. United States
3. Canada

8.

image 搭配 source 屬性

沒有 closing tag

alt 屬性:可以加上敘述, 在 URL 掛掉時說明圖片是甚麼

9 <video src="URL" width="xxx" height="xxx" controls> ... </video>

需要 closing tag !!!!

\*前三屬性應不用說明

\*controls 屬性 表示瀏覽器會提供 開始 暫停 跳過 等影片操作

\*content 可以填入若影片無法載入的文字(類似 alt)

10<a href="URL"> ... </a>

anchor element

超連結用

通常搭配 href 使用

content 可以填超連結想要呈現的名字, 如 This Is A Link To Wikipedia

\*target:可搭配 target 屬性, target="xxx" xxx 有四種值 \_blank(常用, 開 link 於新網頁)

其他\_self、\_parent、\_top()

target 決定超連結如何開啟

\*搭配 img

Content 的地方改成 img tag 可變為圖片連結

11.<table>..</table>

建立表格

參考 table 元素細解

## ===== <head> 使用 =====

### 1.<head>

放入網頁的資訊(給瀏覽器)

### 2.<title> ~ </title>

寫入此網頁的 『網頁名』

### 3.<style>~</style>

內部選取設定 CSS

### 4.<link href="xxx" type="xxx" rel="xxx">

連結外部檔 (如 CSS、JS)

## C. Table element

### 1.<table>~</table>

會建立一表格

中間放置各種 table 元素

### 2.<tr>~</tr>

table row

中間加入 td 以放入資料

th 以決定行、列標題

### 3.<td>~</td>

table data

加在 tr 之中

content 可放置資料

\*span 屬性

可以使用屬性 colspan="n"(or rowspan="n")

合併 n 格格子

### 4.<th scope="XXX">~</th>

table heading

scope 屬性可填入 row、col 決定標題方向

通常會加開一 tr 放置一 th for col

### 5.bolder 屬性 (棄用)

現在多使用 CSS 去設定

```
table, td {  
  border: 1px solid black;  
}
```

### 6<thead> & <tbody> & <tfoot>

類似 HTML 本身的結構，為了管理 table 自然也有 thead 與 tbody

(以直式表格為例)

**thead** 通常用來放置<th scope="col"> 的<tr>

(也就是表格用來放各行代表名的資料)

**tbody** 用來放置剩下所有資料

**tfooter** 用來放總和之類的最後一行

7.善用 CSS 去改善 table 呈現 (參考 CSS 筆記)

當然也可以善用屬性

如 **table** 有屬性 **align**

可以填數 **center right left .....**

8.table 結構

```
<table>
  <thead>
    <tr>
      <th scope="col"> ... </th> <!-- can add more th here-->
    <tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">...</th> <!-- one th for row heading -->
      <td>..</td>
      <td>..</td> <!-- can add more data here -->
    <tr>

    <!-- can add more tr just like above here -->

  </tbody>

  <!-- optional below,if you have footer row can add below -->
  <tfooter>
    <tr>
      <th scope="row">...</th> <!-- one th for row heading -->
      <td>..</td>
      <td>..</td> <!-- can add more data here -->
    <tr>
  <tfooter>
</table>
```

D.form 元素詳解

前言: 淺解網路與通訊協定

\*網路可理解為很多電腦互相交換(接與送)資料

\*HTTP(HyperText Transfer Protocol)規定溝通時的格式, TCP(Transmission Control Protocol)則規定溝通時的傳輸(manage the channel)

\*當輸入一 URL(Uniform Resource Locator), 就建立了一 client-server 的 TCP channel 了  
且 browser 發送 HTTP get, 直到得到 response。

\*URL== 協定名稱://IP(透過 DNS 轉換)

eg:http://nccu.edu

(更詳細參考 HTTP 筆記)

整理:form 當 input submit 被按下時, 會傳出 name="value"的 pair 給 action 指定的程式接收

1.<form>~</form>

表單, 可用於接受資訊

要用屬性告訴 browser 兩件事

(1).資訊要送去哪(action) (通常要傳給 php、js 之類的動態語言才能接收, 此例中 html 是無法接收的) <<參

考JS 如何接收表單<https://pjchender.blogspot.com/2015/11/javascript.html> >>

(2).用HTTP 的甚麼方法(method="POST")

例子如下:

```
<form action="/example.html" method="POST">
</form>
```

當然也可以在 form tag 中加入 child element

2.<input>

輸入

沒有 closing tag

有兩個重要屬性, 一個常用屬性

(1).type

決定接受的 data type 或方式(後面都變 GUI 了)

類型

a."text":文字

b."submit" :送出紐, value 屬性可填入按鈕顯示名, 預設為 submit

c.其他:參考 inputGUI

(2)name

一定要有此屬性, form 送出(submmit)時才會送出此 input !!

eg:

```
<input type="text" name="first-text-field">
```

(3)value:

可以預設 input

例如:value="hello world"

則一開始 input 的 box 會預設 hello world

\*當 form 送出時, 接收方會收到 name=value 的 pair

3.<label for="inputID">~ </label>

\*label 通常用於描述 input、select 在做甚麼 (寫在 content 中)

\*for 屬性填入 input 標籤的 id(所以 input 那邊也要加 id 屬性)

讓兩者連結

(與單純<p>不同就在於label 與input 有連結)

#### 4.inputGUI

(1)input type 的□

a. "password" :密碼使用 (輸出時變\* 或圈圈)

b. "number" :數字 (可加入 step="n"屬性, 增加差距為n 的arrow)  
(打非法字元不會顯示, 至於不合規則的如何處理?)



c. "range" 拉桿 並需要 min、max、step 屬性



d. "checkbox"

```
<label for="pepperoni">Pepperoni</label>
<input id="pepperoni" name="topping" type="checkbox"
value="pepperoni">
```

- \*重點一:選項的name 都要一樣
- \*重點二:check box 因只選要或不要, 所以 Value 一定要預設好
- \*重點三:value 預設並不會顯示, 所以需要label 顯示
- \*重點四:可複選

e. "radio"

just like checkbox 但只能單選

(2).<select>~</select> & <option>~</option>

可以視為input list

```
<form>
  <label for="lunch">What's for lunch?</label>
  <select id="lunch" name="lunch">
    <option value="pizza">Pizza</option>
    <option value="curry">Curry</option>
    <option value="salad">Salad</option>
    <option value="ramen">Ramen</option>
    <option value="tacos">Tacos</option>
  </select>
</form>
```



(3) <datalist> ~<datalist> 有點太偏 GUI 這裡省略

```
<form>
  <label for="city">Ideal city to visit?</label>
  <input type="text" list="cities" id="city"
name="city">

  <datalist id="cities">
    <option value="New York City"></option>
    <option value="Tokyo"></option>
    <option value="Barcelona"></option>
    <option value="Mexico City"></option>
    <option value="Melbourne"></option>
    <option value="Other"></option>
  </datalist>
</form>
```

(4) <textarea id name row cols>~</textarea>

```
<form>
  <label for="blog">New Blog Post: </label>
  <br>
  <textarea id="blog" name="blog" rows="5" cols="30">
  </textarea>
</form>
```

可接收大量文字區 (討論區回應那種)

\*預設文字可寫在content 中

5. 驗證input(valid issue)

有很多方法, 甚至可以使用到regex (正規表達式), 但這篇主要研究html, 就暫不贅述

參考:[https://www.codecademy.com/courses/learn-html/lessons/form-validation/exercises/validation-review?action=resume\\_content\\_item](https://www.codecademy.com/courses/learn-html/lessons/form-validation/exercises/validation-review?action=resume_content_item)