

HTML5 應用實務

什麼是HTML5?跟HTML的差別

HTML5是HTML4的升級版，兩者有以下幾點不同：

1. 新增語義化元素：HTML5新增了一些語義化的元素，例如 `<header>` 、 `<footer>` 、 `<nav>` 、 `<section>` 等等，這些元素可以讓開發者更好地描述頁面的結構，使網頁的內容更具有意義。
2. 視訊和音訊支持：HTML5支持將視訊和音訊直接嵌入網頁，而不需要使用第三方插件如Flash等，這樣使得媒體在網頁上的播放更加流暢和方便。
3. 新的表單屬性：HTML5增加了一些新的表單屬性，例如 `placeholder` 、 `required` 、 `pattern` 等等，使得表單的驗證和填寫更加方便。
4. 新的API支持：HTML5引入了許多新的API，例如地理定位、離線應用、拖放等等，這些API可以幫助開發者更好地實現互動和豐富的網頁體驗。
5. 通用標準：HTML5將以往的各种HTML標準整合成一個通用的標準，這樣使得開發者更容易理解和使用HTML，同時也減少了瀏覽器之間的兼容性問題。

常用HTML5語意標籤

當前HTML5最大的改進之一是導入了語義化標籤，這些標籤有助於開發者在網頁中使用更具有意義的標籤描述其內容，從而提高網頁的可讀性和可維護性。下面是HTML5常用的語意化標籤，以及它們的用途和範例：

HEADER

<header> 標籤通常用於定義一個網頁或者區域的頂部，通常包括標題、導航、搜索框等等。

```
<header>
  <h1>這是網頁標題</h1>
  <nav>
    <ul>
      <li><a href="#">導航一</a></li>
      <li><a href="#">導航二</a></li>
      <li><a href="#">導航三</a></li>
    </ul>
  </nav>
</header>
```

NAV

<nav> 標籤通常用於定義一個網頁或區域的導航。

```
<nav>
  <ul>
    <li><a href="#">導航一</a></li>
    <li><a href="#">導航二</a></li>
    <li><a href="#">導航三</a></li>
  </ul>
</nav>
```

SECTION

<section> 標籤通常用於將網頁分成不同的區塊，每個區塊可以包含一個標題和內容。

```
<section>
  <h2>這是區塊標題</h2>
  <p>這是區塊內容...</p>
</section>
```

ARTICLE

<article> 標籤通常用於定義一篇文章、一個博客帖子或者其他可以獨立存在的內容。

```
<article>
  <h2>這是文章標題</h2>
  <p>這是文章內容...</p>
</article>
```

ASIDE

`<aside>` 標籤通常用於定義一個網頁或區塊的旁邊內容，通常是一些相關資訊或廣告。

```
<aside>
  <h3>廣告</h3>
  <p>最新的產品推銷...</p>
</aside>
```

FOOTER

`<footer>` 標籤通常用於定義一個網頁或區塊的底部標題，可以包含版權訊息、聯繫方式等等。

```
<footer>
  <p>版權所有 © 2023 我的網站</p>
  <p>聯繫我們：info@mywebsite.com</p>
</footer>
```

更多HTML5標籤可至 **W3school手冊**

(<https://www.w3school.com.cn/tags/index.asp>) **查詢**

新的input標籤類型

HTML5引入了許多新的`<input>`標籤類型，以便為各種表單輸入提供更好的支持。以下是一些新的`<input>`類型：

- color：用於選擇顏色的輸入框。
- date：用於選擇日期的輸入框。
- datetime-local：用於選擇日期和時間的輸入框，不包含時區信息。
- email：用於輸入電子郵件地址的輸入框。
- month：用於選擇月份的輸入框。
- number：用於輸入數字的輸入框。
- range：用於選擇範圍內的數字的滑塊。
- search：用於輸入搜索查詢的輸入框。
- tel：用於輸入電話號碼的輸入框。
- time：用於選擇時間的輸入框。
- url：用於輸入網址的輸入框。
- week：用於選擇一周的輸入框。

範例:

```
1  <!-- 顏色選擇器 -->
2  <input type="color" id="colorPicker" name="colorPicker">
3
4  <!-- 日期選擇器 -->
5  <input type="date" id="datePicker" name="datePicker">
6
7  <!-- 電子郵件地址 -->
8  <input type="email" id="email" name="email" placeholder="name@example.com">
9
10 <!-- 數字輸入 -->
11 <input type="number" id="quantity" name="quantity" min="0" max="100">
12
13 <!-- 範圍滑塊 -->
14 <input type="range" id="range" name="range" min="0" max="100" step="1">
```

可參考: **HTML輸入類型** (https://www.w3school.com.cn/html5/att_input_type.asp)

HTML5還引入了一些新的屬性，以改善表單的功能和用戶體驗。以下是一些新的屬性：

1. **autocomplete**：表示瀏覽器是否應顯示自動填充建議。
2. **autofocus**：表示頁面加載時是否自動將焦點設置在此元素上。
3. **form**：將輸入元素與一個或多個表單關聯。
4. **list**：將輸入元素與 `<datalist>` 元素關聯，以提供預定義的選項列表。
5. **max** 和 **min**：分別指定輸入元素可接受的最大和最小值。
6. **multiple**：表示輸入元素

練習: 開啟input_type.html，調整input屬性

媒體相關標籤

1. **<audio>** (<https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/audio>)：用來嵌入聲音文件，支援多種音訊格式，例如MP3、WAV、OGG等。

範例:

```
1 <audio controls>
2   <source src="media/playful-140946.mp3" type="audio/mpeg">
3   Your browser does not support the audio tag.
4 </audio>
```

2. **<video>** (<https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/video>)：用來嵌入影片文件，支援多種視訊格式，例如MP4、WebM等

範例：

```
1 <video controls width="320" height="240">
2   <source src="media/sakura.mp4" type="video/mp4">
3   <source src="movie.webm" type="video/webm">
4   Your browser does not support the video tag.
5 </video>
```

3. **<source>**：用來定義 **<audio>** 和 **<video>** 的媒體來源。

4. **<track>** (<https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/track>)：用來定義 **<video>** 和 **<audio>** 的字幕和章節標題

範例：

```
1 <video controls>
2   <source src="movie.mp4" type="video/mp4">
3   <track src="subtitles_en.vtt" kind="captions" srclang="en" label="English">
4   <track src="subtitles_zh.vtt" kind="captions" srclang="zh" label="Chinese">
5   Your browser does not support the video tag.
6 </video>
```

說明：

當使用者的瀏覽器設定為英文時，字幕將顯示英文字幕檔案 "subtitles_en.vtt"；當使用者的瀏覽器設定為中文時，字幕將顯示中文字幕檔案 "subtitles_zh.vtt"。這是因為在 **<track>** 標籤中，**srclang** 屬性指定了字幕檔案的語言。當瀏覽器偵測到使用者的語言環境和 **srclang** 屬性值相符時，就會自動載入對應的字幕檔案。

如果影片中沒有任何音軌或者音軌中沒有對應的聲音，字幕就不會自動顯示。如果希望在沒有聲音的情況下也能顯示字幕，可以使用 **default** 屬性設置一個預設的字幕檔案

練習：開啟01_Basic/media_tag.html練習媒體標籤

6. `<picture>`：用來適應不同裝置和解析度的圖片，可以使用 `<source>` 來定義不同尺寸和分辨率的圖片

範例:

```
1 <picture>
2   <source srcset="image-large.jpg" media="(min-width: 1200px)">
3   <source srcset="image-medium.jpg" media="(min-width: 768px)">
4   <source srcset="image-small.jpg">
5   
6 </picture>
```

練習: 開啟[01_Basic/picture_tag.html](#)**練習圖片標籤**

7. `<object>`：可以用來嵌入其他類型的媒體檔案，例如 PDF 文件等

範例:

```
1 <object data="file.pdf" type="application/pdf" width="100%" height="800px">
2   <p>Alternative text - include a link
3     <a href="file.pdf">to the PDF!</a>
4   </p>
5 </object>
```

練習: 開啟[01_Basic/object_tag.html](#)**練習媒體標籤**

圖形相關的標籤

1. `<canvas>`：可以用來繪製2D圖形，例如圖形、矩形、直線、文字等

範例:

```
1 <canvas id="myCanvas" width="200" height="100"></canvas>
2
3 <script>
4   var canvas = document.getElementById("myCanvas");
5   var ctx = canvas.getContext("2d");
6   ctx.fillStyle = "#FF0000";
7   ctx.fillRect(0, 0, 150, 75);
8 </script>
```

2. `<svg>`：可以用來繪製向量圖形，可以繪製各種形狀，例如矩形、圓形、直線、曲線等

範例:

```
1 <svg width="100" height="100">
2   <rect x="10" y="10" width="80" height="80" style="fill:blue;stroke:pink;s
3 </svg>
```

3. `<meter>`：可以用來顯示度量值，例如進度條、溫度計等
範例:

```
1 <meter value="75" min="0" max="100">75%</meter>
```

4. `<progress>`：可以用來顯示進度條
範例:

```
1 <progress value="50" max="100"></progress>
```

練習: 開啟01_Basic/html5_tag.html練習媒體標籤

MicroData(微資料)

Microdata 是一種在網頁上標記資料的 HTML 擴充語言，它能夠讓開發人員在網頁中嵌入機器可讀的語義資訊，以協助搜尋引擎、軟體代理人等自動化程式讀取並分析網頁內容。這些標記資料通常是關於產品、地方、人物、事件等等，並且是使用網際網路上的共通協定來表示，例如 **Schema.org**

(<http://Schema.org>), °

以下是一個使用 Microdata 標記資料的範例，這個範例描述了一個產品的資訊，包括產品名稱、價格、評價等等：

```
1 <div itemscope itemtype="http://schema.org/Product">
2   <h1 itemprop="name">Product name</h1>
3   
4   <p itemprop="description">Product description</p>
5   <span itemprop="brand">Brand name</span>
6   <span itemprop="offers" itemscope itemtype="http://schema.org/Offer">
7     <span itemprop="price">19.99</span>
8     <meta itemprop="priceCurrency" content="USD" />
9     <link itemprop="availability" href="http://schema.org/InStock" />
10  </span>
11  <span itemprop="aggregateRating" itemscope itemtype="http://schema.org/Ag
12    <span itemprop="ratingValue">4.5</span>
13    <span itemprop="reviewCount">10</span>
14  </span>
15 </div>
```

在這個範例中，`<div>` 元素的 `itemprop` 屬性被設置為 "Product"，表示這個元素描述了一個產品。在

`<h1>`、``、`<p>` 和 `` 元素中，`itemprop` 屬性被用來標記相應的內容，例如產品的名稱、圖片、描述和品牌。在另外一個 `` 元素中，`itemprop` 屬性被用來標記產品的價格和可用性，並使用 `meta` 元素來指定貨幣。在最後一個 `` 元素中，`itemprop` 屬性被用來標記產品的評價和評價數量。

當 Google 搜尋引擎讀取這個網頁時，它可以根據這些 Microdata 標記來理解這個網頁的內容，例如這個網頁描述了一個產品，並提供了該產品的名稱、圖片、描述、品牌、價格和評價等資訊。

Microdata跟google搜尋有什麼關係

Microdata 與 Google 搜尋有很密切的關係，因為 Google 支援使用 Microdata 標記資料，讓網站擁有者能夠更容易地提供豐富的搜尋結果。使用 Microdata 標記資料，可以讓 Google 明確了解網站內容的含意和關聯性，進而讓 Google 在搜尋結果中顯示更多的資訊，包括價格、評價、詳細描述等等。

例如，當您在 Google 搜尋一個產品時，如果相關網站使用了 Microdata 標記資料，Google 就可以顯示出產品的價格、評價等相關資訊。這不僅能夠讓搜尋結果更加豐富，也能夠提高網站的點擊率和轉換率。

除了 Microdata，Google 也支援其他類型的標記資料，例如 RDFa、JSON-LD 等，它們都是為了幫助開發人員在網站中嵌入機器可讀的語義資訊，進而提高搜尋引擎對網站內容的理解能力，從而顯示更豐富的搜尋結果。

因此，使用 Microdata 或其他類型的標記資料，可以讓您的網站在 Google 搜尋中顯示更豐富的資訊，提高網站的曝光率和流量，進而促進業務發展。

完成標記資料後，可使用 **複合式搜尋結果測試工具**

(<https://search.google.com/test/rich-results>) 測試

練習: 開啟 `microdata.html`，試著在這個網頁中加入 **Microdata** 標記資料，描述一個餐廳的資訊，包括餐廳名稱、地址、營業時間和菜單等等

HTML5 本地儲存和離線應用

HTML5 還提供了本地儲存和離線應用的功能，可以讓 Web 開發者更方便地儲存和訪問用戶的數據，以及在離線狀態下繼續使用應用程式。以下是一些常用的 HTML5 本地儲存和離線應用技術：

1. 本地儲存

HTML5 提供了 Web Storage API，可以讓開發人員在瀏覽器端儲存資料，包括 `localStorage` 和 `sessionStorage` 兩種儲存方式。`localStorage` 和 `sessionStorage` 的主要區別在於資料的生命週期，`localStorage` 的資料在使用者關閉瀏覽器後仍會保留，而 `sessionStorage` 的資料則僅在使用者當前的瀏覽器窗口或標籤頁有效。

以下是一個使用 `localStorage` 儲存資料的範例：

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Local Storage Example</title>
6 </head>
7 <body>
8   <input type="text" id="myInput">
9   <button onclick="saveData()">Save</button>
10  <script>
11    function saveData() {
12      var data = document.getElementById("myInput").value;
13      localStorage.setItem("myData", data);
14      alert("Data saved.");
15    }
16
17    var savedData = localStorage.getItem("myData");
18    if (savedData) {
19      alert("Saved data: " + savedData);
20    }
21  </script>
22 </body>
23 </html>
24
```

在這個範例中，使用者在輸入框中輸入資料後，按下按鈕會使用 `localStorage` 將資料儲存到瀏覽器中。當下次使用者打開這個網頁時，程式會從 `localStorage` 中讀取資料並顯示在提示框中。

2. 離線應用

HTML5 提供了 `Application Cache API`，可以讓開發人員創建離線應用程式，讓使用者在沒有網路連接的情況下仍然能夠訪問網頁。

以下是一個使用 `Application Cache` 離線應用程式的範例：

```
1 <!DOCTYPE html>
2 <html manifest="example.appcache">
3 <head>
4   <title>Example App</title>
5 </head>
6 <body>
7   <h1>Example App</h1>
8   <p>This is an example of an offline web application.</p>
9 </body>
10 </html>
```

當瀏覽器無法連接到網路，它會自動使用本地儲存的資源文件，讓使用者仍然能夠訪問網頁。

以下是一個 `example.appcache` 文件的範例：

```
1  CACHE MANIFEST
2
3  # Version 1.0.0
4
5  CACHE:
6  index.html
7  style.css
8  script.js
9  image.jpg
10
11 NETWORK:
12 http://code.jquery.com/jquery-1.11.2.min.js
13
14 FALLBACK:
15 / fallback.html
16
```

說明:

1. CACHE 區塊包含一個或多個 URI，表示需要快取的檔案。在下面的例子中，我們快取了 index.html 和一個名為 styles.css 的樣式表。
2. NETWORK 區塊包含一個或多個 URI，表示不應該被快取的檔案。在下面的例子中，我們不希望快取 jQuery 函式庫。
3. FALLBACK 區塊包含兩個 URI，表示當使用者離線時，應該提供替代資料的 URI。在下面的例子中，我們希望在離線時顯示 fallback.html 的內容。

當載入html檔案時，瀏覽器將檢查 manifest 屬性，下載清單檔案，並快取清單中列出的檔案。當使用者離線時，瀏覽器將使用快取中的檔案，並將 Fallback 指令轉向指定的 URI。

HTML5 表單的驗證和提交應用

HTML5 表單提供了一些內置的驗證機制，可以使用 required 屬性設置必填，也可以使用其他屬性進行驗證，例如：

```
minlength：最小長度。
maxlength：最大長度。
min：最小值。
max：最大值。
pattern：正規表達式。
```

以下是一個使用了驗證屬性的表單範例：

```
1  <form>
2    <label for="name">姓名 : </label>
3    <input type="text" id="name" name="name" required minlength="2" maxlength:
4    <br>
5    <label for="age">年齡 : </label>
6    <input type="number" id="age" name="age" min="18" max="60">
7    <br>
8    <label for="email">電子郵件 : </label>
9    <input type="email" id="email" name="email" required pattern="[a-zA-Z0-9.
10   <br>
11   <input type="submit" value="提交">
12 </form>
```

在表單驗證失敗時，HTML5 表單會自動顯示錯誤提示。
其中電子郵件採用正規表達式來驗證

驗證電子郵件格式的正規表達式 (regular expression)

(https://developer.mozilla.org/zh-TW/docs/Web/JavaScript/Guide/Regular_Expressions) °

具體而言，它將電子郵件地址拆分為三個部分：使用者名稱、域名和頂級域名。並且使用一些特定字符和模式來確定它們的格式和長度。

以下是這段正規表達式的拆分說明：

1. `[a-zA-Z0-9.%±]+`: 這個部分用來匹配使用者名稱。它要求使用者名稱必須由至少一個英文字母、數字、點 (.)、下劃線 ()、百分號 (%)、加號 (+) 或減號 (-) 組成。加號和減號用於支持某些特殊的電子郵件服務商 (例如Google Mail) 的地址格式。
2. `@`: 這個符號是使用者名稱和域名之間的分隔符。
`[a-zA-Z0-9.-]+`: 這個部分用來匹配域名。它要求域名必須由至少一個英文字母、數字、點 (.) 或破折號 (-) 組成。破折號用於支持某些特殊的電子郵件服務商 (例如Outlook) 的地址格式。
`.`: 這是一個特殊的字符序列，用來匹配域名和頂級域名之間的句點。
3. `[a-zA-Z]{2,}`: 這個部分用來匹配頂級域名。它要求頂級域名必須由至少兩個英文字母組成。這將避免匹配像 ".com" 或 ".edu" 這樣的簡短字符串，而只匹配像 ".com.au" 或 ".edu.cn" 這樣的完整頂級域名。 **

這段正規表達式要求電子郵件地址必須符合常見的格式，例如 "**user@example.com** (<mailto:user@example.com>)"，並且必須包含使用者名稱、域名和頂級域名三個部分。如果電子郵件地址不符合這些標準，它將不會通過驗證。

透過js進行表單驗證:

```
1 <form onsubmit="validateForm()">
2   <div class="mb-3">
3     <label for="name" class="form-label">姓名 :</label>
4     <input type="text" class="form-control" id="name" name="name">
5     <div class="invalid-feedback">請輸入您的姓名。</div>
6   </div>
7   <div class="mb-3">
8     <label for="age" class="form-label">年齡 :</label>
9     <input type="number" class="form-control" id="age" name="age">
10    <div class="invalid-feedback">請輸入您的年齡。</div>
11  </div>
12  <div class="mb-3">
13    <label for="email" class="form-label">電子郵件 :</label>
14    <input type="email" class="form-control" id="email" name="email">
15    <div class="invalid-feedback">請輸入有效的電子郵件地址。</div>
16  </div>
17  <button type="submit" class="btn btn-primary">提交</button>
18 </form>
19
20
21 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
22 <script>
23 function validateForm() {
24   // 使用 jQuery 獲取表單元素的值
25   var name = $("#name").val();
26   var age = $("#age").val();
27   var email = $("#email").val();
28
29   if (name == "") {
30     alert("請輸入姓名");
31     return false;
32   }
33
34   if (age < 18 || age > 60) {
35     alert("年齡必須在18到60之間");
36     return false;
37   }
38
39   if (!email.match(/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}/)) {
40     alert("請輸入正確的電子郵件地址");
41     return false;
42   }
43
44   return true;
45 }
46
47 function submitForm() {
48   var form = $('#myForm')[0];
49   var formData = new FormData(form);
50
51   $.ajax({
52     type: "POST",
53     url: "submit.php",
54     data: formData,
55     contentType: false,
56     processData: false,
57     success: function(data) {
58       alert("表單提交成功!");
59     },
60     error: function(jqXHR, textStatus, errorThrown) {
61       alert("表單提交失敗: " + textStatus);
62     }
63   });
64 }
65 </script>
```