

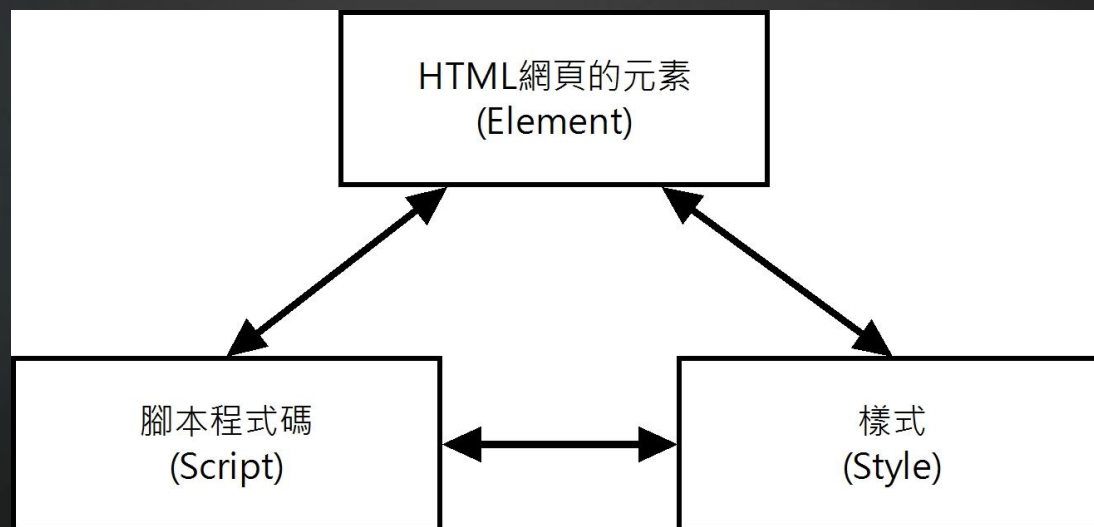
DOM物件模型

DOM物件模型的基礎

- ▶ 物件模型與DOM
- ▶ DOM基礎的HTML網頁內容

物件模型與DOM-說明

- ▶ 「物件模型」(Object Model) 對於HTML網頁來說，就是一種規範如何存取HTML元素、樣式或程式碼的機制，可以將HTML元素、樣式和程式碼視為物件，和定義之間的關係，如下圖所示：



物件模型與DOM-簡介

- ▶ DOM物件模型提供一組標準程式介面，可以讓我們透過這組介面來存取物件的屬性和方法，換句話說，程式設計者可以使用此程式介面來瀏覽HTML網頁或XML文件，也可以新增、刪除和修改節點資料。對於HTML網頁來說，DOM主要是由兩大部分組成，如下所示：
 - ▶ DOM Core：提供HTML網頁或XML文件瀏覽、處理和維護階層架構，主要提供兩種功能，如下所示：
 - ▶ 瀏覽（Navigator）：能夠在網頁的樹狀架構中走訪節點。
 - ▶ 參考（Reference）：能夠存取節點的集合物件。
 - ▶ DOM HTML：HTML網頁專屬的DOM API介面，其目的是將網頁元素都視為是一個個物件，以便讓JavaScript等程式語言存取元素來建立動態網頁內容。

物件模型與DOM-規格

- ▶ DOM Level 0：這並不是真正存在的規格，它是建立DOM Level 1規格的基礎，目前瀏覽器支援Level 0的主要目的只是為了和舊版相容。
- ▶ DOM Level 1：在1998年成為W3C（World Wide Web Consortium）的建議規格，提供HTML網頁和XML文件通用的瀏覽和處理元素內容的物件模型，不過，在Level 1並沒有提供事件處理機制。
- ▶ DOM Level 2：在2000年成為W3C的建議規格，除了DOM2 Core和DOM2 HTML外，還支援事件處理機制的DOM2 Events等6項規格。
- ▶ DOM Level 3：在2004年成為W3C的建議規格，Level 3包含5種規格，即DOM3 Core、Load and Save、Validation、Events和XPath。

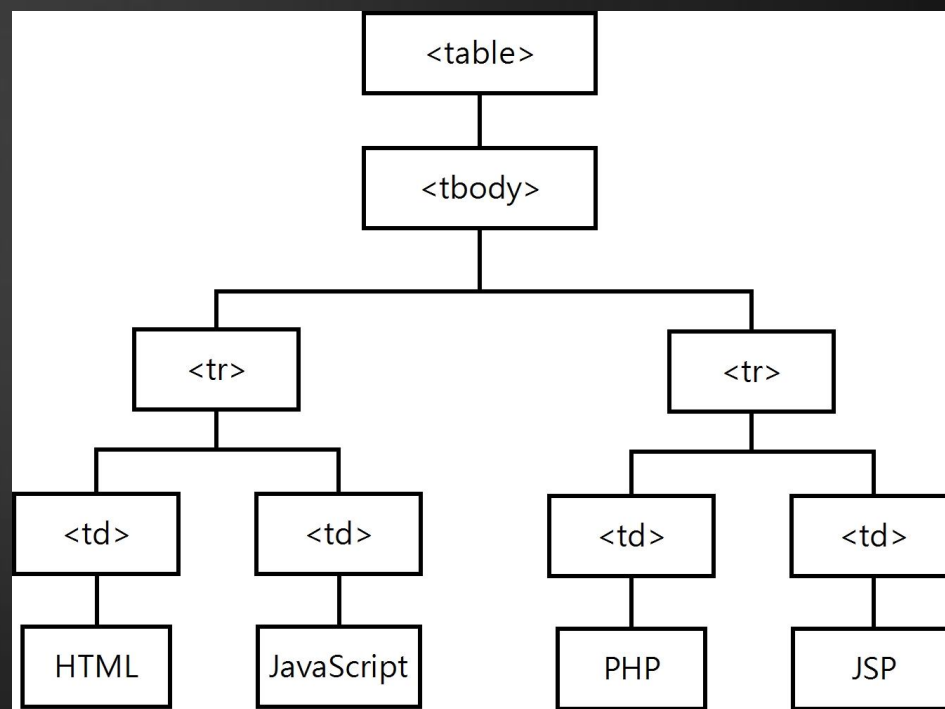
物件模型與DOM-優點

- ▶ 提供跨平台和程式語言的程式介面：DOM提供應用程式標準的程式處理介面，這是一種HTML和XML文件的標準API。
- ▶ 支援多種文件：DOM支援HTML網頁和XML文件，其中DOM Core可以使用在HTML網頁和XML文件；DOM HTML是針對HTML網頁。
- ▶ 支援多種程式語言：DOM支援多種程式語言和腳本語言，例如：Java、PHP、.NET語言和ECMAScript（即JavaScript）等。

DOM基礎的HTML網頁內容

- DOM是一種結構化文件內容的物件模型，可以將HTML網頁的標籤和文字內容視為「節點」（Node），依各節點之間的關係連接成樹狀結構，例如：
<table>表格標籤片斷，如下所示：

```
<table>  
  <tbody>  
    <tr>  
      <td>HTML</td>  
      <td>JavaScript</td>  
    </tr>  
    <tr>  
      <td>PHP</td>  
      <td>JSP</td>  
    </tr>  
  </tbody>  
</table>
```



取得HTML元素節點

使用Id屬性取得元素節點

► getElementById()

_DOM的getElementById()方法可以取出HTML網頁指定的HTML元素，和傳回節點物件的參考，它是使用參數的id屬性值來取得指定的元素，如下所示：

```
var a = document.getElementById("google");
```

- 程式碼的document可以取得HTML網頁的DOM，方法的參數是<a>標籤的id屬性值，換句話說，就是取回此a元素的節點物件，如下所示：

```
<p><a id="google" href="http://www.google.com.tw">Google</a></p>
```

使用**標籤名稱**取得元素節點

▶ nodeName與tagName屬性

- ▶ 在取得指定節點物件後，我們可以使用nodeName或tagName屬性取得節點的標籤名稱，傳回的是大寫的標籤名稱字串，如下所示：

```
alert(a.nodeName + " - " + a.href);  
alert(btn.tagName + " - " + btn.type);
```

- ▶ 程式碼不只可以傳回標籤名稱，在之後的href和type屬性就是HTML標籤的原生屬性，在取得指定節點物件後，就可以存取這些屬性值。

使用class名稱取得元素節點

- ▶ 當我們要擷取網頁當中某個class元件時，可採用

```
var x = document.getElementsByClassName("intro");
```

```
<script>
function getClass()
{
    var cl=document.getElementsByClassName("items");
    alert(cl.length);
}
</script>
```

A. 英雄聯盟成員：

1. 雷神索爾
2. 浩克

B. 迪士尼的卡通人物：

1. 米奇
2. 雪寶

class



JavaScript 通知

4

確定

取得元素節點 - 範例一

```
<style>
  body{font-size: 18px;font-weight: bolder;color:blue;line-height: 200%}
</style>
<script>
function changeTextStyle()
{
  var myText=document.getElementById("myText");
  myText.style.borderColor='black';
  myText.style.borderStyle="dotted";
  myText.style.borderWidth="0 0 1px 0 ";
}

function showText()
{
  var myName=document.getElementById("myText")
}

function showElement()
{
  var a=document.getElementById("myText");
  document.write("節點種類 :"+a.nodeName + "<br/> 節點資料內容 : "+a.value);
}

</script>
</head>
<body onload="changeTextStyle()">
  Input Your Name : <input type="text" value="" id="myText"/> <br/>
  <input type="button" value="送出" id="myBtn" onclick="showElement()">
</body>
```

Input Your Name : JoLin

送出



節點種類 :INPUT
節點資料內容 : JoLin

mod_getNode_Id1.html

使用Id屬性取得元素節點-範例二

```
<script>
function showAElement() {
    var a = document.getElementById("google");
    alert(a.nodeName + " - " + a.href);
}
function showButtonElement() {
    var btn = document.getElementById("test");
    alert(btn.tagName + " - " + btn.type);
}
</script>
</head>
<body>
<h2>使用Id屬性取得元素節點</h2>
<hr/>
<p id="content">使用Id屬性取得元素節點</p>
<p><a id="google"
    href="http://www.google.com.tw">Google</a></p>
<input id="test" type="button"
    onclick="showAElement()" value="選取<a>元素">
<input type="button"
    onclick="showButtonElement()"
    value="選取<input>元素">
</body>
</html>
```



使用標籤名稱取得元素節點-TagName

▶ getElementsByTagName()方法

- ▶ DOM的getElementsByTagName()方法可以傳回指定標籤名稱的節點陣列或清單，它是一個NodeList物件，如下所示：

```
var p = document.getElementsByTagName("p");
```

- ▶ 程式碼取回HTML網頁中所有<p>標籤的節點，因為可能有多個同名標籤，所以傳回的不是單一節點物件，而是NodeList物件，我們可以使用length屬性取得共有幾個節點物件，如下所示：

```
alert("P元素有: " + p.length + "個");
```

- ▶ 程式碼可以取得HTML網頁共有幾個<p>標籤。

使用標籤名稱取得元素節點-item()

- ▶ 使用item()方法取得指定的節點物件
- ▶ `getElementsByTagName()`方法傳回的是一個NodeList物件，我們可以使用item()方法來取得指定的節點物件，如下所示：

```
var f = document.getElementsByTagName("input");  
for (var i = 0; i < f.length; i++) {  
    objEle = f.item(i);  
    strTags += objEle.type + " ";  
}
```

- ▶ 程式碼取得<input>標籤的NodeList物件後，使用for迴圈一一取出節點物件，迴圈是從0至f.length-1，如下所示：

```
objEle = f.item(i);
```

使用標籤名稱取得元素節點-索引

- ▶ 使用陣列索引方式取得節點物件

- ▶ 除了使用item()方法外，我們也可以直接將NodeList物件當成陣列來取出指定索引值的節點物件，如下所示：

```
var f=document.getElementsByTagName("input");  
for (var i = 0; i < f.length; i++)  
strTags += f[i].type + " ";
```

- ▶ 程式碼和前述for迴圈的功能相同，只是改為陣列索引來取得節點物件，即f[i]。

取得元素節點-範例

```
<script>
function getEle1() {
    var p = document.getElementsByTagName("p");
    alert("P元素有: " + p.length + "個");
}
function getEle2() {
    var h = document.getElementsByTagName("h2");
    alert("H2元素有: " + h.length + "個");
}
function showItems1() {
    var strTags = "";
    var f = document.getElementsByTagName("input");
    for (var i = 0; i < f.length; i++) {
        objEle = f.item(i);
        strTags += objEle.type + " ";
    }
    alert(strTags);
}
function showItems2() {
    var strTags = "";
    var f = document.getElementsByTagName("input");
    for (var i = 0; i < f.length; i++)
        strTags += f[i].type + " ";
    alert(strTags);
}
</script>
```

HTML元素H2

HTML元素P

HTML元素P

HTML元素H2

取得P元素 取得H2元素 顯示INPUT元素 顯示INPUT元素

練習一 取得元素內容

- ▶ 請設計一個網頁如圖所示
- ▶ 當按鍵按下去後，則會顯示相關訊息如圖所示
- ▶ 提示：
 - ▶ 這個題目需要一個迴圈來進行判斷
 - ▶ 若要辨識物件的種類可應用 物件名稱.Type

Input Your Name :

Input Your Phone Number :

Input Your ID Number :

Input Your Web Site:



按下按鈕後

這個網頁的Input元件有:5 個
這個網頁的Button元件有:1 個
這個網頁的label元件有:0 個

取得與更改元素內容

- ▶ 在取得指定的DOM節點物件後，我們可以使用innerHTML屬性存取HTML元素的文字內容，這是HTML 5版支援的屬性，如下所示：

```
var c =  
document.getElementById("content");  
alert(c.innerHTML);  
c.innerHTML = "JavaScript和DOM";
```

- ▶ 程式碼在取得id屬性值為"content"的節點物件後，使用innerHTML屬性取得此HTML元素的文字內容，然後再一次使用innerHTML屬性，更改HTML元素的文字內容。

取得與更改元素內容-範例

```
<script>
function showText() {
    var li = document.getElementById("myLI");
    alert(li.innerHTML);
}
function changeText() {
    var c = document.getElementById("content");
    alert(c.innerHTML);
    c.innerHTML = "JavaScript和DOM";
}
</script>
</head>
<body>
<h2>取得與更改元素內容</h2>
<hr/>
<p id="content">取得與更改元素內容</p>
<ul id="myUL">
    <li>HTML</li>
    <li id="myLI">JavaScript</li>
    <li>VBScript</li>
    <li>PHP</li>
    <li>XML</li>
</ul>
<input type="button" onclick="showText()" value="顯示元素內容">
<input type="button" onclick="changeText()" value="更改元素內容">
</body>
```



練習二 取得元素內容

- ▶ 延續上一個練習，當按鈕按下後，請顯示使用者輸入的資訊
- ▶ 請注意，在`document.write`的元素加入後，迴圈會遭到中斷，因此建議要顯示的文字，可以宣告一個變數存放，等元素取得完畢後，再一次顯示即可

Input Your Name :

Input Your Phone Number :

Input Your ID Number :

Input Your Web Site:



User Name : JoLin
User Phone Number : 0988123456
User ID Number : A1234567889
User Web Site : WWW.NPUST.EDU.TW

HTML集合物件

HTML集合物件

- ▶ DOM提供一些屬性來取得常用HTML元素的集合物件，其說明如下所示：
 - ▶ **document.anchors**屬性：包含HTML網頁所有擁有name屬性的<a>標籤。
 - ▶ **document.forms**屬性：包含HTML網頁所有<form>標籤。
 - ▶ **document.images**屬性：包含HTML網頁所有標籤。
 - ▶ **document.links**屬性：包含HTML網頁所有擁有href屬性的<a>標籤。

HTML集合物件-範例(以<a>及<frame>物件為例)

```
<script>
function showForms() {
    var f = document.forms;
    alert("<FORM>: " + f.length);
}
function showLinks() {
    var links = document.links;
    var strTags = "<A> : " + links.length + "\n";
    for (var i = 0; i < links.length; i++) {
        objEle = links.item(i);
        strTags += objEle.href + "\n";
    }
    alert(strTags);
}
function showAnchors() {
    var a = document.anchors;
    var strTags = "<A> : " + a.length + "\n";
    for (var i = 0; i < a.length; i++) {
        objEle = a[i];
        strTags += objEle.href + "\n";
    }
    alert(strTags);
}
</script>
```

HTML集合物件

[Google HiNet](#) Test

表單集合

超連結集合1

超連結集合2

存取HTML標籤的屬性



- ▶ 在取得HTML網頁的節點物件後，我們不只可以存取元素內容，也可以存取標籤屬性，如下所示：

```
var p = document.getElementsByTagName("p");  
alert(p.item(0).align);
```

- ▶ 程式碼取得所有p元素後，就可以取得第1個<p>標籤的align屬性值，同樣方式，我們可以存取HTML標籤的其他屬性。在DOM節點物件提供3種方法來處理屬性，其說明如下所示：

方法	說明
getAttribute(attribute)	取得傳入參數attribute屬性的屬性值
setAttribute(attribute,value)	將傳入參數attribute屬性設定為value值
RemoveAttribute(attribute)	刪除傳入參數的attribute屬性

存取HTML標籤的屬性-範例

```
<script>
function setAlign(strAlign) {
    var p = document.getElementById("myP");
    p.align = strAlign;
}
function getAlign() {
    var p = document.getElementsByTagName("p");
    alert(p.item(0).align);
}
function switchAlign() {
    var objEle = document.getElementsByTagName("p");
    if (objEle[1].getAttribute("align") == "left")
        objEle[1].setAttribute("align", "center");
    else
        objEle[1].setAttribute("align", "left");
}
</script>
```

存取HTML標籤的屬性

存取HTML標籤的屬性1

存取HTML標籤的屬性2

置中對齊

取得標籤屬性

切換標籤屬性



按下置中對齊後

存取HTML標籤的屬性

存取HTML標籤的屬性1

存取HTML標籤的屬性2

置中對齊

取得標籤屬性

切換標籤屬性

取得標籤屬性

- ▶ 取得網頁當中HTML物件後，不只可存取元素內容，也可以取得標籤屬性

```
var p=document.getElementsByTagName("p");  
alert(p.item(0).align);
```

以上兩行的範例，第一行代表的是取得P元素
第二行為取得P元素後，開始取得align的屬性

```
<html>  
<head>  
  <meta charset="utf-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <title>取得標籤屬性</title>  
  <script>  
    function getAlign()  
    {  
      var p=document.getElementsByTagName("p");  
      alert("第一行"+p.item(0).align+"\n 第二行"+p.item(1).align+"\n 第三行"+p.item(2).align);  
    }  
  </script>  
</head>  
<body>  
  <p align="left">我是第一行(靠左對齊); </p>  
  <p align="right">我是第二行(靠右對齊); </p>  
  <p align="center">我是第二行(置中對齊); </p>  
  <input type="button" name="btn1" value="對齊屬性" onclick="getAlign()>  
</body>  
</html>
```

我是第一行(靠左對齊);

我是第二行(靠右對齊);

我是第二行(置中對齊);

對齊屬性



JavaScript 通知

第一行left
第二行right
第三行center

確定

練習三-取得標籤屬性

- ▶ 請將範例 `mod_getAttribute.html`
- ▶ 請設計一個函數，內容為例用迴圈的方式，取得各"p"元件的對齊的方式，並存成字串，回傳給呼叫這個函數的程式
- ▶ 請將 `alert("第一行"+p.item(0).align+"\n 第二行"+p.item(1).align+"\n 第三行"+p.item(2).align);` 這一行利用呼叫函數的方式，顯示內容

DOM的節點操作

插入和新增節點-步驟一：建立節點物件

- ▶ 首先使用Document物件的createElement()方法建立節點物件，如果標籤擁有文字節點，還需要使用createTextNode()方法建立文字節點物件，如下所示：

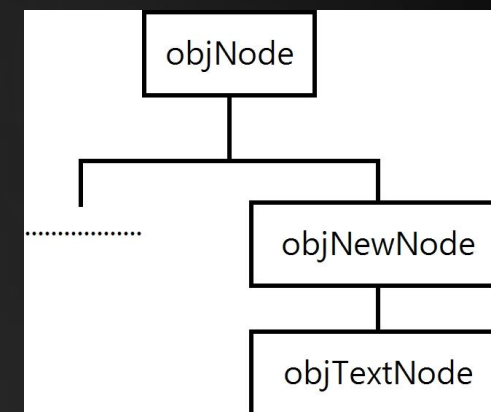
```
var objNewNode = document.createElement("P");  
var objTextNode = document.createTextNode('新增段落');
```

- ▶ 程式碼建立<p>標籤的節點，參數的標籤名稱字串為大寫，在<p>標籤擁有文字內容，所以建立文字節點，參數為文字節點的內容。

插入和新增節點-步驟二將節點物件新增或插入DOM節點樹

- ▶ 在建立節點物件後，就可以將節點物件新增或插入DOM節點樹，如下所示：

```
objNode.appendChild(objNewNode);  
objNewNode.appendChild(objTextNode);
```



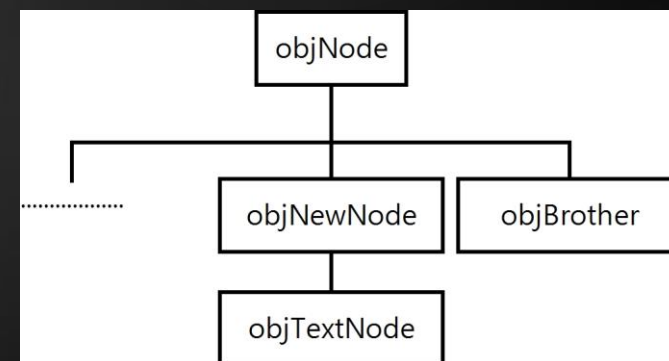
- ▶ 程式碼的objNode為DOM節點樹的節點，首先將objNewNode節點物件新增為objNode節點物件的最後一個子節點，然後將文字節點objTextNode新增為objNewNode的子節點。

插入和新增節點-插入節點

- ▶ 我們也可以將節點物件插入節點樹中指定兄弟節點之前，如下所示：

```
objNode.insertBefore(objNewNode, objBrother);  
objNewNode.appendChild(objTextNode);
```

- ▶ 程式碼的objNode和objBrother為DOM節點樹的節點，首先將objNewNode節點插入objNode子節點objBrother兄弟節點之前，然後將文字節點objTextNode新增為objNewNode的子節點。



插入和新增節點-範例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>插入與新增節點</title>
<script>
function appendNode() {
    var objNode = document.getElementById("myDiv");
    var strText = "新增段落";
    var objNewNode = document.createElement("p");
    var objTextNode = document.createTextNode(strText);
    objNode.appendChild(objNewNode);
    objNewNode.appendChild(objTextNode);
}
function insertNode(objNode, objBrother, strText) {
    var objNode = document.getElementById("myDiv");
    var objBrother = objNode.children[1];
    var strText = '段落一.二';
    var objNewNode = document.createElement("p");
    var objTextNode = document.createTextNode(strText);
    objNode.insertBefore(objNewNode, objBrother);
    objNewNode.appendChild(objTextNode);
}
</script>
</head>
<body id="myBody">
<h2>在DOM節點樹插入和新增節點</h2>
<hr/>
<div id="myDiv">
<p onclick="appendNode()">段落一</p>
<p onclick="insertNode()">段落二</p>
<p>段落三</p>
</div>
</body>
</html>
```

在DOM節點樹插入和新增節點

段落一

段落二

段落三

段落四



按了三下段落一文字後

在DOM節點樹插入和新增節點

段落一

段落二

段落三

新增段落

新增段落

新增段落

設定插入節點的樣式

- ▶ 若要將插入的節點直接更改顏色

```
var objTextNode = document.createTextNode(strText);  
objNewNode.appendChild(objTextNode);  
objNewNode.style.color="blue";
```

- ▶ 若要直接套用某特定的樣式，可應用以下的設定：

```
var objNewNode = document.createElement("P");  
objNewNode.appendChild(objTextNode);  
objNewNode.className="newStyle";
```

更改插入節點樣式範例

```
<script>
function appendNode() {
    var objNode = document.getElementById("myDiv");
    var strText = "新增段落";
    var objNewNode = document.createElement("P");
    var objTextNode = document.createTextNode(strText);
    objNode.appendChild(objNewNode);
    objNewNode.appendChild(objTextNode);
    objNewNode.style.color="blue";
}
function insertNode(objNode, objBrother, strText) {
    var objNode = document.getElementById("myDiv");
    var objBrother = objNode.children[1];
    var strText = '段落一.二';
    var objNewNode = document.createElement("P");
    var objTextNode = document.createTextNode(strText);
    objNode.insertBefore(objNewNode, objBrother);
    objNewNode.appendChild(objTextNode);
    objNewNode.className="newStyle";
}
</script>
</head>
<body id="myBody">
<h2>在DOM節點樹插入和新增節點</h2>
<hr/>
<div id="myDiv">
    <p onclick="appendNode()">段落一</p>
    <p onclick="insertNode()">段落二</p>
    <p>段落三</p>
</div>
</body>
```

在DOM節點樹插入和新增節點

段落一

段落二

段落三



在DOM節點樹插入和新增節點

段落一

段落一.二

段落一.二

段落二

段落三

新增段落

新增段落

刪除節點-刪除

- ▶ 在JavaScript程式碼可以使用DOM節點物件的removeChild()方法刪除節點樹中指定的子節點，如下所示：

```
objParent.removeChild(objNode);
```

- ▶ objParent是父節點物件，removeChild()方法可以刪除參數的子節點，即objNode。

刪除節點-顯示子節點

- ▶ 本節程式範例是使用printChilds()函數來顯示指定節點的所有子節點，if條件是使用hasChildNodes()方法檢查節點是否擁有子節點，如下所示：

```
if (objNode.hasChildNodes()) {  
    var nodeCount = objNode.children.length;  
    strMsg += "子元素數 = " + nodeCount + "\n";  
    for (var i = 0; i < nodeCount; i++)  
        strMsg += "標籤名稱 = " + objNode.children[i].nodeName + "\n";  
    alert(strMsg);  
}
```

刪除節點-範例

```
<script>
function printChilds(objNode) {
    var strMsg = "節點名稱 =" + objNode.nodeName + "\n";
    if (objNode.hasChildNodes()) {
        var nodeCount = objNode.children.length;
        strMsg += "子元素數 =" + nodeCount + "\n";
        for (var i = 0; i < nodeCount; i++)
            strMsg += "標籤名稱 =" + objNode.children[i].nodeName + "\n";
        alert(strMsg);
    }
}
function deleteChild(objParent, objNode) {
    var strMsg = "刪除標籤 =" + objNode.nodeName + "\n";
    strMsg += "刪除標籤的子節點數 =" + objNode.children.length + "\n";
    objParent.removeChild(objNode);
    alert(strMsg);
}
</script>
```

```
<body id="myBody">
<h2>刪除DOM節點樹的節點</h2>
<hr/>
<div id="myDiv">
<p><span>段落一</span></p>
<p id="myP2"><span>段落二</span></p>
<p id="myP3"><span>段落三</span></p>
</div>
<form>
<script>
var d = document.getElementById("myDiv");
var p2 = document.getElementById("myP2");
var p3 = document.getElementById("myP3");
</script>
<input type="button" value="顯示子節點" onclick="printChilds(d);">
<input type="button" value="刪除元素1" onclick="deleteChild(d, p2);">
<input type="button" value="刪除元素2" onclick="deleteChild(d, p3);">
</form>
</body>
```

刪除DOM節點樹的節點

段落一

段落二

段落三

顯示子節點

刪除元素1

刪除元素2



顯示子節點

應用程式 ★ Bookmarks Download Page Java H

刪除DOM節點樹的節點

段落一

段落二

段落三

顯示子節點 刪除元素1 刪除元素2

JavaScript 通知

節點名稱 = DIV
子元素數 = 3
標籤名稱 = P
標籤名稱 = P
標籤名稱 = P

確定

刪除節點-範例二

- ▶ 當我們有多個節點要一次刪除時，我們可以應用以下的方式
 - ▶ 步驟一 先獲得要刪除節點的父節點名稱
 - ▶ 步驟二 設定變數，獲取子節點的陣列
 - ▶ 步驟三 利用迴圈，刪除子節點

```
function deleteallChild()  
{  
    var node=document.getElementById("myDiv");  
    var childs = node.childNodes;  
    for(var i = childs.length - 1; i >= 0; i--)  
    {  
        node.removeChild(childs[i]);  
    }  
}
```

刪除DOM節點樹的節點

段落一

段落二

段落三

顯示子節點

刪除元素1

刪除元素2

刪除所有元素



刪除DOM節點樹的節點

顯示子節點

刪除元素1

刪除元素2

刪除所有元素

練習四-新增刪除節點(很重要)

- ▶ 請開啟題目檔mod_adddeleteNode_Q.html
- ▶ 新增的項目請套用css樣式newItems
- ▶ 請建立四個函式其功能如下
 - ▶ 函式一：用來判斷使用者按下哪個按鈕並根據判斷結果，呼叫函式二顯示資訊
 - ▶ 函式二：將Input Box裡面所輸入的內容，append到指定項目下
 - ▶ 函式三:按下刪除鍵，則刪除題目一下所有的答案項目
 - ▶ 函式四 送出資料後，請將input box裡面的資料清除 (請應用 `obj.value=""` 來清空);

A. 英雄聯盟成員：

1. 雷神索爾
2. 浩克

B. 迪士尼的卡通人物：

1. 米奇
2. 雪寶



A. 英雄聯盟成員：

1. 雷神索爾
2. 浩克
3. 鋼鐵人

B. 迪士尼的卡通人物：

1. 米奇
2. 雪寶

鋼鐵人

A. 英雄聯盟成員：

B. 迪士尼的卡通人物：

1. 米奇
2. 雪寶

以下請自行Study

DOM的節點瀏覽

DOM的節點瀏覽

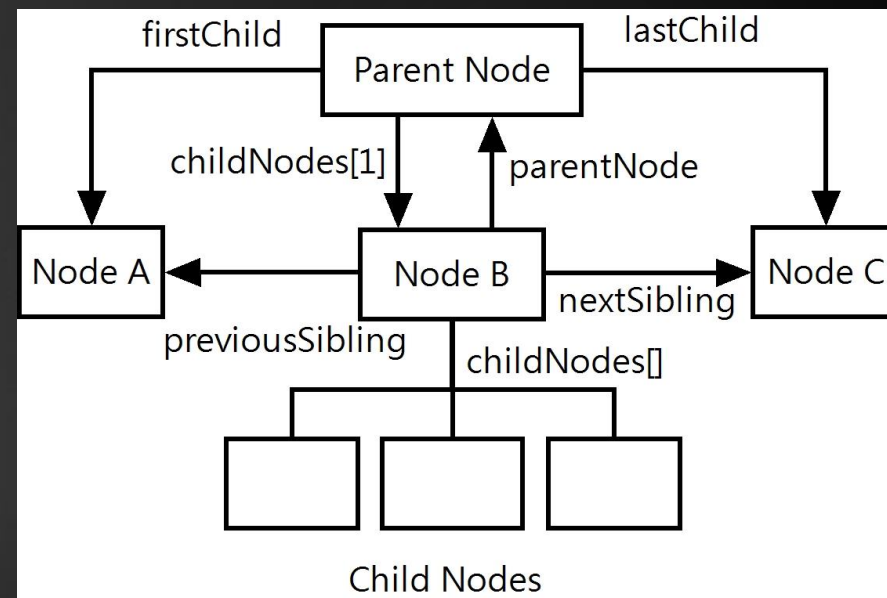
- ▶ DOM瀏覽節點的相關屬性
- ▶ 瀏覽父節點
- ▶ 瀏覽兄弟節點
- ▶ 瀏覽子節點與子元素

DOM瀏覽節點的相關屬性-瀏覽

屬性	說明
firstChild	傳回第1個childNodes集合物件的子節點，包含此節點下的所有子節點
lastChild	傳回最後1個childNodes集合物件的子節點，包含此節點下的所有子節點
parentNode	傳回父節點的物件
nextSibling	傳回下一個兄弟節點的物件
previousSibling	傳回前一個兄弟節點的物件
nodeName	傳回節點的HTML標籤名稱，名稱是英文大寫字母，例如：P和TABLE
nodeType	傳回節點種類，1為標籤；2為屬性；3為文字節點
specified	傳回HTML標籤是否擁有屬性的布林值

DOM瀏覽節點的相關屬性-圖例

- ▶ Parent Node使用firstChild屬性取得第1個子節點Node A，childNodes[1]為第2個子節點Node B，childNodes是所有子節點的集合物件，lastChild屬性可以取得最後一個子節點Node C。
- ▶ 在Node B使用previousSibling屬性可以取得前一個兄弟節點Node A；nextSibling屬性取得下一個兄弟節點Node C。取得上一層父節點是使用parentNode屬性。



DOM瀏覽節點的相關屬性-存取

- ▶ DOM提供存取屬性來存取文字節點的內容，其說明如下所示：
 - ▶ data屬性：存取文字節點的內容，其他種類的節點傳回undefined。
 - ▶ nodeValue屬性：存取文字節點的內容，其他種類的節點傳回null。

DOM瀏覽節點的相關屬性-集合物件

- ▶ DOM的集合物件可以取得下一層子節點和節點屬性，其說明如下所示：
 - ▶ `attributes`屬性：節點屬性的集合物件，可以直接使用名稱來存取。
 - ▶ `childNodes`屬性：子節點的集合物件，這是一個陣列物件，可以使用從0開始的索引值進行存取，例如：`childNodes[0]`。
 - ▶ `children`屬性：子元素的集合物件，不包含文字節點的子節點，這也是一個陣列物件，可以使用從0開始的索引值進行存取，例如：`children[0]`。

瀏覽父節點

- ▶ 在DOM取得父節點是使用parentNode屬性，例如：項目符號的HTML標籤和，li的父節點是ul，如下所示：

```
var objLi = document.getElementById("myLI");  
var objNode = objLi.parentNode;  
alert(objNode.nodeName);
```

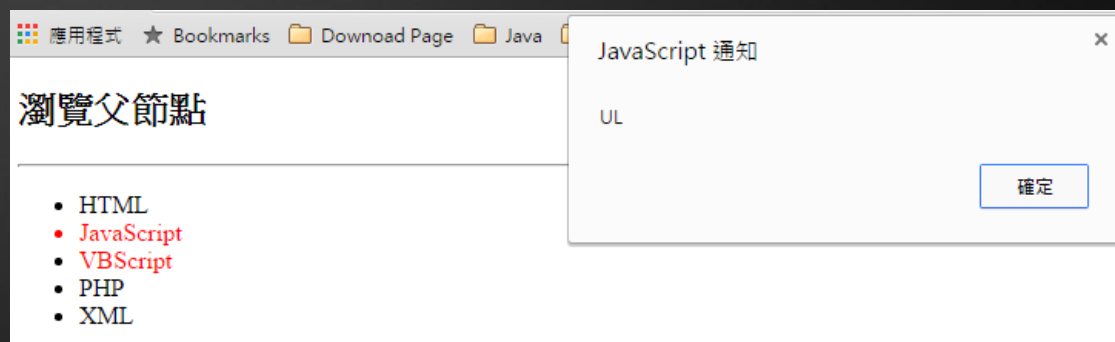
- ▶ 程式碼的myLI為標籤的id屬性值，可以使用parentNode屬性取得上一層HTML標籤名稱UL，使用的是nodeName屬性。

瀏覽父節點-範例

```
<script>
function showParentNode() {
    var objLi = document.getElementById("myLI");
    var objNode = objLi.parentNode;
    alert(objNode.nodeName);
}
function showParentNode1() {
    var objSpan = document.getElementById("mySpan");
    var objNode = objSpan.parentNode;
    alert(objNode.nodeName);
}
</script>
</head>
<body>
<h2>瀏覽父節點</h2>
<hr/>
<ul id="myUL">
    <li>HTML</li>
    <li style="color:red" id="myLI"
        onclick="showParentNode()">JavaScript</li>
    <li><span style="color:red" id="mySpan"
        onclick="showParentNode1()">VBScript</span></li>
    <li>PHP</li>
    <li>XML</li>
</ul>
</body>
```

瀏覽父節點

- HTML
- JavaScript
- VBScript
- PHP
- XML



瀏覽兄弟節點

- ▶ DOM可以使用previousSibling和nextSibling屬性取得兄弟節點，筆者同樣是使用清單項目的HTML標籤和為例，如下所示：

```
var objLi = document.getElementById("myLI");  
var objNode = objLi.previousSibling;  
alert(objNode.firstChild.data);
```

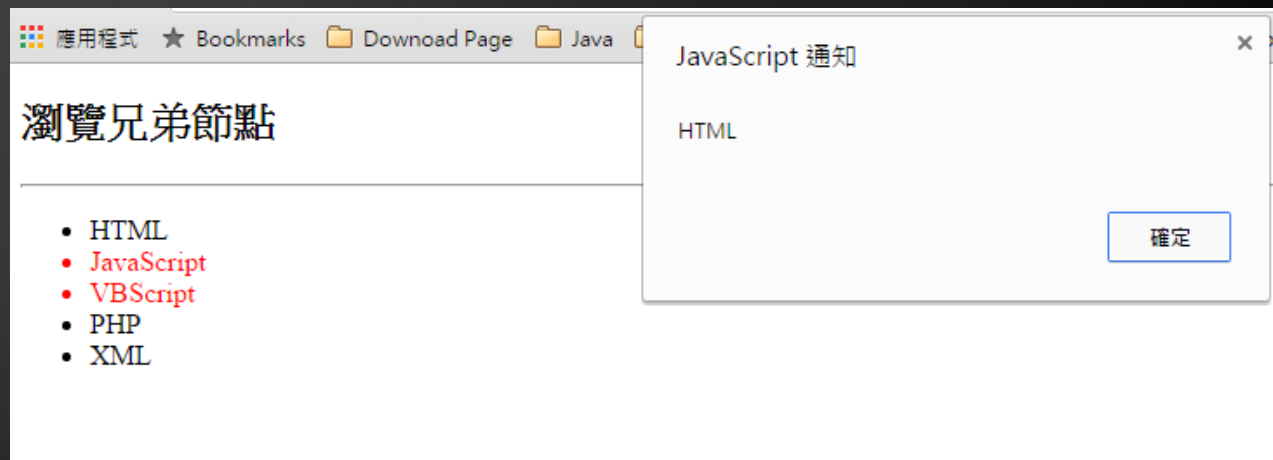
- ▶ 程式碼的myLI是標籤物件，可以取得前一個兄弟節點，然後使用firstChild屬性取得下一層文字節點的data屬性（也可以直接使用innerHTML屬性取代）。

瀏覽兄弟節點-範例

```
<script>
function showNode1() {
    var objLi = document.getElementById("myLI");
    var objNode = objLi.previousSibling;
    alert(objNode.firstChild.data);
}
function showNode2() {
    var objLi = document.getElementById("myLI1");
    var objNode = objLi.previousSibling;
    alert(objNode.innerHTML);
}
</script>
</head>
<body>
<h2>瀏覽兄弟節點</h2>
<hr/>
<ul id="myUL">
    <li>HTML
    <li style="color:red" id="myLI"
        onclick="showNode1()">JavaScript
    <li style="color:red" id="myLI1"
        onclick="showNode2()">VBScript
    <li>PHP
    <li>XML
</ul>
</body>
```

瀏覽兄弟節點

- HTML
- JavaScript
- VBScript
- PHP
- XML



瀏覽子節點與子元素-取得

- ▶ DOM節點可以使用childNodes屬性取得子節點的集合物件，這是一個陣列物件，可以使用從0開始的索引值進行存取。首先取得父節點的節點物件，如下所示：

```
var objUL = document.getElementById("myUL");
```

- ▶ 程式碼取得id屬性值myUL的ul元素後，我們就可以使用childNodes屬性取得所有子節點，如下所示：

```
var objChlds = objUL.childNodes;
```

- ▶ 程式碼可以取得ul元素的所有子節點，包含li元素和文字節點。

瀏覽子節點與子元素-瀏覽

- ▶ 接著就可以使用for迴圈一一取得每一個節點物件且將節點名稱顯示出來，如下所示：

```
var strTags = "";  
for (var i = 0; i < objChlds.length; i++) {  
    objEle = objChlds.item(i);  
    strTags += objEle.nodeName + "\n";  
}
```

- ▶ 程式碼是使用childNodes屬性，如果使用children屬性，取得的子節點就只有li子元素，而沒有文字節點。

瀏覽子節點與子元素-範例

```
<script>
function showChilds1() {
    var objUL = document.getElementById("myUL");
    var objChilds = objUL.childNodes;
    var strTags = "";
    for (var i = 0; i < objChilds.length; i++) {
        objEle = objChilds.item(i);
        strTags += objEle.nodeName + "\n";
    }
    alert(strTags);
}
function showChild2() {
    var objUL = document.getElementById("myUL");
    var objChilds = objUL.children;
    var strTags = "";
    for (var i = 0; i < objChilds.length; i++) {
        objEle = objChilds[i];
        strTags += objEle.nodeName + "\n";
    }
    alert(strTags);
}
</script>
</head>
<body>
<h2>瀏覽子節點與子元素</h2>
<hr/>
<ul id="myUL">
    <li>HTML</li>
    <li style="color:red" id="myLI"
        onclick="showChilds1()">JavaScript</li>
    <li style="color:red" id="myLI1"
        onclick="showChild2()">VBScript</li>
    <li>PHP</li>
    <li>XML</li>
</ul>
</body>
```

