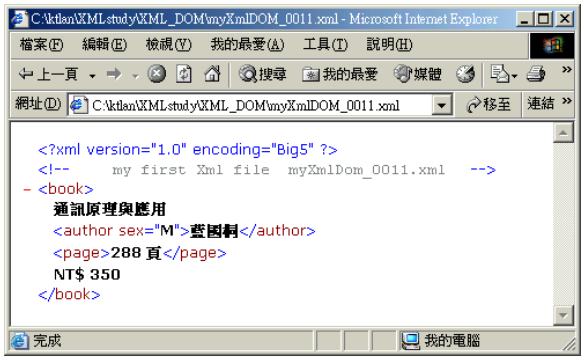
Lecture 1: 使用檔名：myXmlDOM_0010.xml & myXmlDOM_0010.html

一、先 edit 一個 myXmlDOM_0010.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>
<!--    my first Xml file    myXmlDom_0010.xml -->

<book>
        <title>通訊原理與應用</title>
        <author sex="M">藍國桐</author>
        <page>288  頁</page>
        <price>NT$ 350</price>
</book>
```



二、再 edit 一個 myXmlDOM_0010.html 檔 。

```
<Html>
    <Head>
        <Title>Book Description </Title>
          <script Language="JavaScript" FOR="window" EVENT="onload">
            var oDOM=dsoBook.XMLDocument;

            myInfo.innerText=oDOM.documentElement.legnth;      //documentElement 不是一個 nodeList
            myTitle.innerText=oDOM.documentElement.text;       // documentElement 是 DOM 的東東
            myAuthor.innerText=oDOM.documentElement.childNodes(0).text;      //text 是 MicroSoft 的東東
          </script>
    </Head>

    <Body>
        <XML ID="dsoBook" SRC="myXmlDOM_0010.xml"></XML>
        <H2> Book Description </H2>

        <Span style="color:#0000FF">Xml file information: </Span>
        <Span ID="myInfo"></Span> <br>

         <Span style="color:#0000FF">Title: </Span>
```

It is a MicroSoft's **DataIsland**!!

```
        <Span ID="myTitle"></Span> <br>

        <Span style="color:#0000FF">Author: </Span>
        <Span ID="myAuthor"></Span> <br>

    </Body>
</Html>
```

說明一、使用 Data Island 的方式將 myXmlDOM_0010.xml 載進來，並形成一個 Document Object Model (DOM) 物件 - oDOM。

Step 1: 形成 Microsoft DataIsland：

```
<XML ID="dsoBook" SRC="myXmlDOM_0010.xml"></XML>
```

Step 2: 在 JavaScript 的 coding 區塊中，宣告一個 DOM 物件：

```
var oDOM=dsoBook.XMLDocument;
```

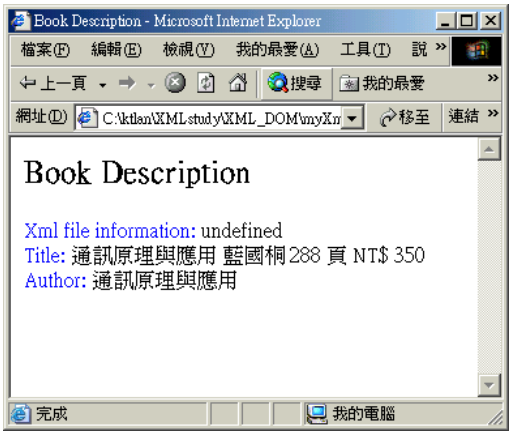
Specifically, the **XMLDocument** member contains the root object of the DOM, known as the *Document node*.
(XMLDocument 指的是“整份文件”而非 Document (root) Element)

說明二、叫用 DOM 物件的方法(method) 及屬性(property)，並填入 HTML 檔中的物件中 (如 SPAN)。

```
myInfo.innerText=oDOM.documentElement.legnth;
myTitle.innerText=oDOM.documentElement.text;
myAuthor.innerText=oDOM.documentElement.childNodes(0).text;
```

DOM 物件 的屬性或是子物件

DOM 物件



說明三、Adding a data island to an HTML page causes Internet Explorer to create both a DSO (represented directly by the data island's ID) and a DOM (accessed through the DSO's **XMLDocument** member).

說明四、叫用 DOM 物件的子物件或是其所提供的方法(method) 並循著樹狀(tree) 的物件結構找出所要找的子物件的屬性(property)。

三、另外再 edit 一個 myXmlDOM_0011.html 檔 。

```
<Html>
    <Head>
        <Title>Book Description </Title>

        <script Language="JavaScript" FOR="window" EVENT="onload">
            var oDOM;
            oDOM=new ActiveXObject("MSXML.DOMDocument");
            oDOM.async=false;
            oDOM.load("myXmlDOM_0010.xml");

            myInfo.innerText=oDOM.documentElement.legnth;     //documentElement 不是一個 nodeList
            myTitle.innerText=oDOM.documentElement.text;          // documentElement 是 DOM 的東東
            myAuthor.innerText=oDOM.documentElement.childNodes(0).text;
        </script>
    </Head>

    <Body>

        <H2> Book Description </H2>
```

載入 XML 檔的三大步驟。

沒有設定 DataIsland

…… (The same as the "myXmlDOM_0010.html")

說明一、利用 ActiveX 物件的載入方式才能在 HTML 檔案中形成 DOM 物件。

說明二、The Structure of the DOM ：。

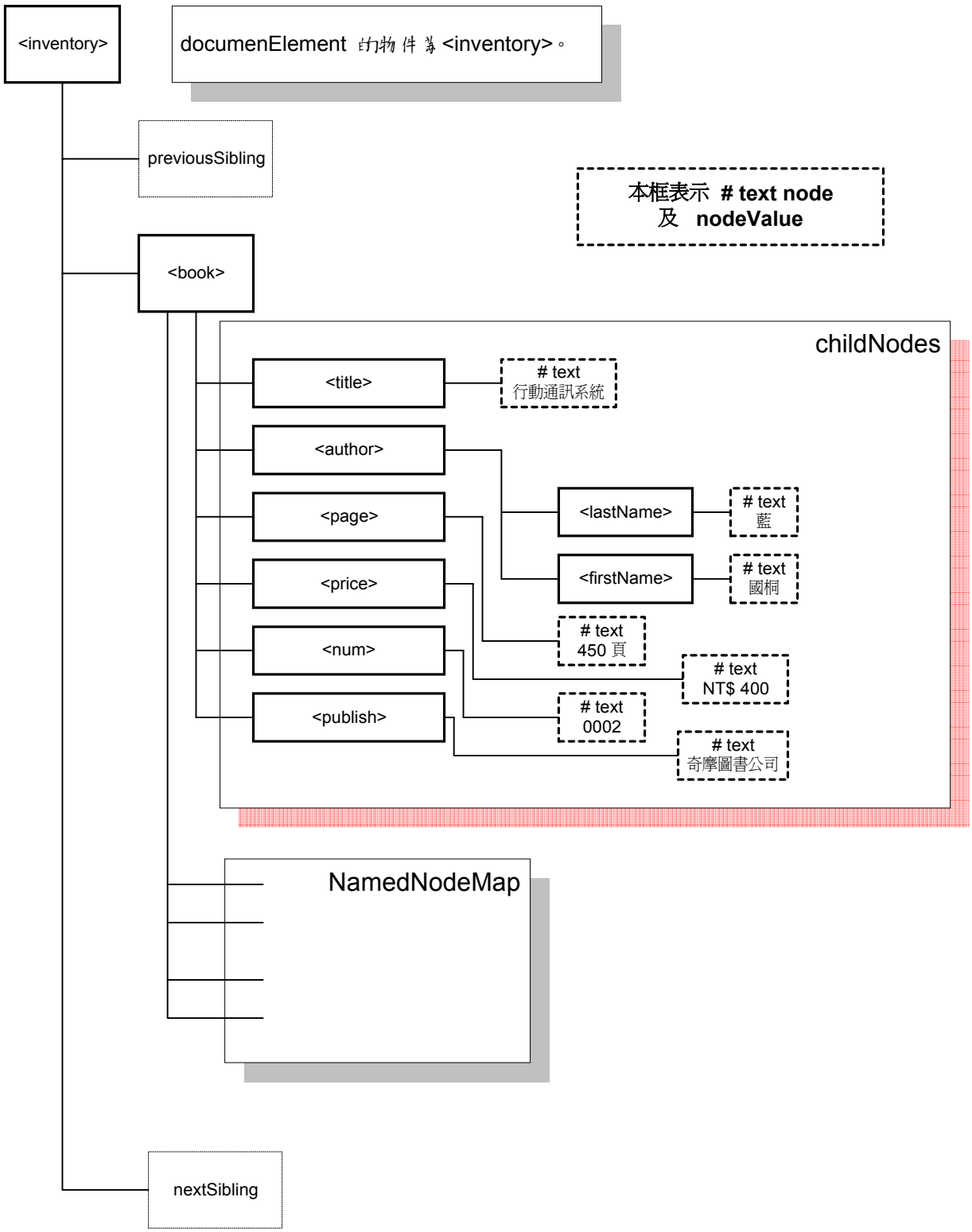
Node type (*nodeType* property) (*nodeTypeString* property)	XML document component that the node object represents	**Node Name** (*nodeName* property)	**Node value** (*nodeValue* property)
Document (9)		#document	*null*
Element (1)		Element type name (ex. author)	*null*
Text (3)		#text	The parent XML component's text (ex. 通訊原理與應用)
attribute (2)		Attribute name (ex. sex)	Attribute value (ex. M)
processingInstruction (7)			The entire content of the processing instruction
comment (8)		#comment	All the text within the comment delimiters (ex. my first Xml file myXmlDom_0010.xml)

以下列 XML 為例：

```
<?xml version="1.0" encoding="Big5" ?>
<!--   my Xml file   myXmlCSS_0080.xml   -->
<inventory>
    <book>
        <title>通訊原理與應用 </title>
        <author>
            <lastName>藍</lastName>
            <firstName>國桐 </firstName>
        </author>
        <page>288  頁 </page>
        <price>NT$ 350</price>
        <num>0001 </num>
        <publish>全華圖書 公司</publish>
    </book>

    <book>
        <title>行動通訊系統</title>
            <author>
            <lastName>藍</lastName>
            <firstName>國桐 </firstName>
        </author>
        <page>450  頁 </page>
        <price>NT$ 400</price>
        <num>0002</num>
        <publish>奇摩圖書 公司</publish>
    </book>
    <book>
        ……
    </book>
    <book>
        ……
    </book>
</inventory>
```

其 DOM 物件結構為：



<book> 是一個 node 也是一個 element 物件，它含有 <title><author><page><price><num><publish> 等子物件(均為 element 物件)。它另含有一個 childNodes 的集合物件，可以用來指向上述子物件。 (只要是一個 node 物件，它就會有一個 childNodes 的集合

物件，不論這個 node 物件是否只含有一個或多個子物件。)
利用 node.childNodes 可以得到 這個 nodeList 物件。

同理，要得到 NamedNodeMap 集合物件可以用 node.Attributes 得到。

檢查 DOM 物件結構的 JavaScript 範例(myXmlDOM__0012.html)：

(請自行 navigating 一遍 XML 物件)

```
// myNodeTest.innerText=oDOM.nodeName;
// myNodeTest.innerText=oDOM.nodeType;
// myNodeTest.innerText=oDOM.nodeTypeString;
// myNodeTest.innerText=oDOM.nodeValue;
// myNodeTest.innerText=oDOM.text;
// myNodeTest.innerText=oDOM.xml

// myNodeTest.innerText=oDOM.firstChild.nodeName;
// myNodeTest.innerText=oDOM.firstChild.nodeType;
// myNodeTest.innerText=oDOM.firstChild.nodeTypeString;
// myNodeTest.innerText=oDOM.firstChild.nodeValue;
// myNodeTest.innerText=oDOM.firstChild.text;
// myNodeTest.innerText=oDOM.firstChild.xml;

// myNodeTest.innerText=oDOM.childNodes(1).nodeName;
// myNodeTest.innerText=oDOM.childNodes(1).nodeType;
// myNodeTest.innerText=oDOM.childNodes(1).nodeTypeString;
// myNodeTest.innerText=oDOM.childNodes(1).nodeValue;
// myNodeTest.innerText=oDOM.childNodes(1).text;
// myNodeTest.innerText=oDOM.childNodes(1).xml;

// myNodeTest.innerText=oDOM.documentElement.nodeName;
// myNodeTest.innerText=oDOM.documentElement.nodeType;
// myNodeTest.innerText=oDOM.documentElement.nodeTypeString;
// myNodeTest.innerText=oDOM.documentElement.nodeValue;
// myNodeTest.innerText=oDOM.documentElement.text;
// myNodeTest.innerText=oDOM.documentElement.xml;
```

```
// myNodeTest.innerText=oDOM.documentElement.childNodes(1).nodeName;
// myNodeTest.innerText=oDOM.documentElement.childNodes(1).nodeType;
// myNodeTest.innerText=oDOM.documentElement.childNodes(1).nodeTypeString;
// myNodeTest.innerText=oDOM.documentElement.childNodes(1).nodeValue;
// myNodeTest.innerText=oDOM.documentElement.childNodes(1).text;
// myNodeTest.innerText=oDOM.documentElement.childNodes(1).xml;


//    myNodeTest.innerText=oDOM.documentElement.childNodes(1).attributes.length;
//    myNodeTest.innerText=oDOM.documentElement.childNodes(1).childNodes.length;
//    myNodeTest.innerText=oDOM.documentElement.childNodes(1).childNodes(0).nodeTypeString;
//    myNodeTest.innerText=oDOM.documentElement.childNodes(1).xml;
```

檢查 DOM 物件結構的 VB6.0 範例(xmlDOMProject01)：

(請自行 navigating 一遍 XML 物件)

```
Dim objDOM As MSXML.DOMDocument
Set objDOM = New MSXML.DOMDocument

objDOM.async = False
blnGetXml = objDOM.Load("c:/ktlan/myXmlDom_0032.xml")
      ' 載入 一個 XML 檔案，成功的話則傳回 true
```

```
List1.AddItem "root nodeName: " & objDOM.nodeName
List1.AddItem "root nodeType: " & objDOM.nodeType
List1.AddItem "node(0) : " & objDOM.childNodes(0).nodeTypeString
List1.AddItem "root nodeValue: " & objDOM.nodeValue
List1.AddItem "node(0) : " & objDOM.childNodes(0).Text
List1.AddItem "node(0) : " & objDOM.childNodes(0).xml
```

VB6.0 範例（myXMLDOM_Project02）：

Lecture 2: 使用檔名：myXmlDOM_0020.xml & myXmlDOM_0020.html
(Navigating a complicated xml file !)

一、edit 一個 myXmlDOM_0020.html 檔 。

(demo)

1. 在 myXmlDOM_0020.html 中使用 myXmlDOM_0011.xml
2. 在 myXmlDOM_0020.html 中使用 myXmlDOM_0020.xml

Lecture 3: 使用檔名：myXmlDOM_0021.xml & myXmlDOM_0021.html
(Navigating the attribute nodes of a complicated xml file !)

一、edit 一個 myXmlDOM_0021.html 檔 。

(demo)

1. 在 myXmlDOM_0021.html 中使用 myXmlDOM_0021.xml

Lecture 4: DOM 的屬性(properties)及方法(methods)

Table 1. Useful common properties provided by all **node** types.

Property	Description
attributes	A **NamedNodeMap** collection of all this node's Attribute child nodes
childNodes	A **NodeList** collection of all this node's non-attribute child nodes
firstChild	This node's first non-attribute child node
lastChild	This node's last non-attribute child node
previousSibling	The previous node at the same level as this node
nextSibling	The following node at the same level as this node
nodeName	This node's name
nodeType	A numeric code indicating the type of this node
nodeTypeString	A string containing the type of this node, in lowercase
nodeValue	The value of this node (似乎只限 text node 及 attribute node 有這個值)
parentNode	The father node of this node
ownerDocument	The root document node of the document containing this node
text	The entire text content of this node and of all descendent element nodes. (MS ?)
xml	The entire XML content of this node and all its descendent nodes. (MS ?)

A property will contain the value *null* if that property doesn't apply to the node. (沒有值)

Table 1-1. Useful common methods provided by all **node** types.

Method	Description
inserBefore (*newChild, refChild*)	Inserts the *newChild* node before the existing *refChild*.
removeChild (*oldChild*)	Removes *oldChild* from the list, and returns it.
appendChild (*newChild*)	Adds *newChild* to the end of the list, and returns it.
haschildNodes ()	Return a Boolean to check if the node has any children.

Table 2. Useful properties and methods provided by **Document** nodes.

Property	Description
async	Assigning false to the async property causes a subsequent call to the load method. (that is, the call won't return until the document fully loaded). (MS ?)
documentElement	The Element node representing the root element of the XML document
parseError	An object that contains information on the first well-formedness or validity error that the processor encountered when it loaded and processed the XML document. (MS ?)
schemas	If you assign an XML SchemaCache object to this property, the document will be validated against the associated XML schema when the load method is called. (MS ?)
url	The URL of the XML document. (MS ?)

Table 2-1. Useful methods provided by **Document** nodes.

Property	Description
getElementsByTagName (*tagname*)	Returns a NodeList collection of all elements in the document that have the specified type name. If you pass "*", it returns all elements. ex. nodeNow=oDOM. GetElementByTagName ("author")
createElement (*tagname*)	Creates an element, with the name specified.
createAttribute (*name*)	Creates an attribute, with the specified *name*.

Table 3. Useful properties provided by **Element** nodes.

Element node property	Description
tagName	The name of the element

Table 3-1. Useful methods provided by **Element** nodes.

Element node method	Description
getAttribute (attr-name)	Returns the value of the element's attribute that has the specified name. Ex. strName= nodeNow.getAttribute ("color")
setAttribute (*name, value*)	Sets the value of the specified attribute to this new value.
removeAttribute (*name*)	Removes the specified attribute.
getAttributeNode (attr-name)	Returns the Attribute node representing the element's attribute that has the specified name. Ex. nodeAttribute= nodeNow.getAttributeNode ("color");
setAttributreNode (newAttr)	Adds a new attribute node.
RemoveAttributeNode (oldAttr)	Removes the specified attribute node.

getElementsByTagName (type-name)	Returns a **NodeList** collection of Element nodes for all of this element's descendent elements that have the specified type name.
	Ex.
	nodeList= nodeNow.getElementsByTagName("title");
	strName= nodeList(1).childNodes(0).nodeValue;

Table 4. The property and methods provided by a **NodeList collection** object.

NodeList property	Description
length	The number of nodes contained in the collection. ex. nodeNow.childNodes.length
item(0-based) (default property)	Returns the node at the position indicated by the index ex. nodeNow.childNodes(i) or nodeNow.childNodes.item(i)
reset()	Sets the internal pointer to the position before the first node in the collection, so the next call to *nextNode* returns the first node. (MS ?)
nextNode()	Returns the next node in the collecion. (MS ?)

Table 5. The properties provided by **NamedNodeMap** collection objects.

NamedNodeMap property	Description
length	The number of nodes contained in the collection
item (0-based-index) (default method)	(MS ?)
reset()	(MS ?)
nextNode()	(MS ?)

Table 5-1. The methods provided by **NamedNodeMap** collection objects.

NamedNodeMap method	Description

getNamedItem(att-name)	Returns the node that has the specified name, or null if the attribute isn't found
	Ex.
	nodeMap=nodeNow.attributes;
	strName=nodeMap.getNamedItem("color");
	document.write(strName.nodeValue);

setNamedItem (arg)	The arg parameter is a Node object, which is added to the list.
removeNamedItem (name)	Removes the Node specified by name

Table 6. A useful property and method provided by **Text** nodes.

Text node property	Description
length	The number of characters in the node's text. (MS ?)
substringData (char-offset, num-chars) *(從第 char-offset 抓 num-chars 個字元)*	Return a string containing the specified number of characters from the node's text content, starting at the specified character offset. (MS ?)
splitText (offset)	Separates this single Text node into two adjacent Text nodes.

Lecture 5: 利用操作 DOM 的屬性(properties)及方法(methods) ，可以增減 XML 的各種資料節點，因此可以在兩個 process 之間傳送合法 (well-formed 及 valid) 的 XML 文件。

(範例：textbook 的 ASP 程式 ch9-5-1.asp)

```
<%@ LANGUAGE="JavaScript"%>
<HTML>
<HEAD>
<TITLE>伺服端顯示 XML 元素</TITLE>
<BODY>
<H2>顯示 XML 文件的元素</H2>
<HR>
<%
// 建立 XML DOM 物件
var xmlDom = Server.CreateObject("MSXML2.DOMDocument.4.0")
xmlDom.async = "false"
// 載入 XML 文件
xmlDom.load(Server.MapPath("Ch9-5.xml"))
if (xmlDom.parseError.errorCode != 0)
    Response.Write("載入 XML 文件錯誤<P>");
else{
    Response.Write("<TABLE border=1>")
    var objListNode = xmlDom.documentElement.childNodes
    // 顯示所有的XML 節點
    for (i = 0 ; i < objListNode.length; i++){
        Response.Write("<TR><TD>" + objListNode.item(i).nodeName + "</TD>")
        Response.Write("<TD>" + objListNode.item(i).text + "</TD></TR>")
    }
    Response.Write("<TABLE>")
}
%>
</BODY>
</HTML>
```

(demo：課本 p.9-40)

先行研習課題：(1) 利用 TCP/IP winsock ，兩個 process 可以互傳字串 (2) 了解 HTTP 的通訊 protocol 。