



網頁程式設計

JavaScript程式設計

Instructor: 馬豪尚

JavaScript物件

- › JavaScript的物件是一種關聯陣列(associative array)
- › 陣列儲存的資料稱為元素，物件儲存的資料為屬性

物件名稱

屬性

```
var user = {name: Kevin; age: 20;}
```

key value key value

JavaScript物件點記法

- › 物件成員的數值可能是任何東西，例如：字串、數字、陣列、甚至是函式
- › 可透過點記法 (Dot notation) 存取物件的屬性與函式
 - 寫一個「點」以及你所想存取的項目，可能是簡單屬性的名稱、陣列屬性的項目，又或是針對物件函式之一的呼叫
- › 其實你一直在使用物件
 - array.push()
 - window.alert()

物件名稱 屬性

user.age



JavaScript基本內建物件

JavaScript內建物件- Number

› Number物件提供了操作數值的屬性與方法

方法	說明
isNaN(x)	判斷傳入的值是不是 NaN.
isFinite(x)	判斷傳入的值是不是一個有限的數值
isInteger(x)	判斷傳入的值是不是一個整數
isSafeInteger(x)	判斷傳入的值是不是在 IEEE-754 雙精度範圍間
parseFloat(x)	傳回將傳入的值轉成浮點數的結果
parseInt(x)	傳回將傳入的值轉成整數的結果
toString(x)	傳回將傳入的值轉成字串的結果

› Example

– Number.isNaN(10)

JavaScript內建物件- String

› String物件提供了操作字串的屬性和方法

方法	說明
indexOf(str[, start])	傳回str參數首次出現在呼叫此方法的字串中的索引，可搭配start參數設定從哪個索引開始搜尋
lastIndexOf(str[, start])	傳回str參數最後出現在呼叫此方法的字串中的索引
includes(str[, start])	傳回字串是否包含str參數
startsWith(str[, start])	傳回字串是否以參數str開頭
endsWith(str[, start])	傳回字串是否以參數str結尾
split(str)	根據參數str做分割字串，分割完的結果以Array物件形式回傳
slice(begin[, end])	傳回索引為begin~(end-1)的字串
substr(index, length)	傳回從索引index開始位置擷取長度為length的字串
concat(str1, ..., strN)	此方法會回傳一個新的字串，由呼叫此方法的字串及作為參數傳入的字串組合而成
repeat(count)	傳回將字串重複count的結果

JavaScript內建物件-Math物件

- › Math物件提供了操作數值運算的屬性與方法
- › Math物件的成員均為靜態屬性和靜態方法，不能使用new關鍵字來建立Math物件

屬性	說明
E	自然數 $e=2.718281828459045$
LN2	e 為底的對數2
LN10	e 為底的對數10
LOG2E	2為底的對數 e
LOG10E	10為底的對數 e
PI	圓周率
SQRT1_2	$\frac{1}{2}$ 的平方根
SQRT2	2的平方根

JavaScript內建物件-Math物件

› 基本運算方法

方法	說明
abs(num)	傳回參數num的絕對值
max(n1, n2, ...)	傳回參數中的最大值
min(n1, n2, ...)	傳回參數中的最小值
pow(n1, n2)	傳回n1的n2次方值
random()	傳回0-1.0之間的一個亂數值
sign(num)	傳回參數的正負符號，1表示正數，-1表示負數，0表示0

JavaScript內建物件-Math物件

› 進位或捨去運算方法

方法	說明
ceil(num)	傳回大於等於參數num的整數
floor(num)	傳回小於等於參數num的整數
round(num)	傳回參數num的四捨五入值
trunc(num)	傳回參數num的整數部分

JavaScript內建物件-Math物件

› 平方根、指數、對數

方法	說明
<code>sqrt(num)</code>	傳回參數num的平方根
<code>cbrt(num)</code>	傳回參數num的立方根
<code>exp(num)</code>	傳回自然數e的num次方
<code>log(num)</code>	傳回參數num以e為底的對數
<code>log10(num)</code>	傳回參數num以10為底的對數
<code>log2(num)</code>	傳回參數num以2為底的對數

JavaScript內建物件-Math物件

› 三角函數

– 參數num為弧度， $\text{弧度} = \text{角度} * \text{PI} / 180$

方法	說明
sin(num)	傳回參數num的正弦值
cos(num)	傳回參數num的餘弦值
tan(num)	傳回參數num的正切值
asin(num)	傳回參數num的反正弦值
acos(num)	傳回參數num的反餘弦值
atan(num)	傳回參數num的反正切值

JavaScript內建物件-Date物件

- › JavaScript 的 Date 物件提供來做跟日期和時間相關的操作
- › 宣告Date物件(現在時間)
 - `var today = new Date();`
- › 宣告特定時間點的 Date 物件
 - 傳入一個數字 (Timestamp)
 - › `var date = new Date(1481361366000);`
 - › 值表示從 1970-01-01 00:00:00 UTC (格林威治標準時間) 開始累計到某時間點的毫秒數 (milliseconds)
 - 傳入多個數字，指定年、月、日、時、分、秒、毫秒
 - › `var birthday = new Date(1990, 1, 17, 23, 30, 15);`

JavaScript內建物件-Date物件

› Date 物件提供的方法 (Methods)

方法	用途
getFullYear()	取得是幾年 (yyyy)
getMonth()	取得是幾月 (0-11)
getDate()	取得是幾日 (1-31)
getHours()	取得是幾時 (0-23)
getMinutes()	取得是幾分 (0-59)
getSeconds()	取得是幾秒 (0-59)
getMilliseconds()	取得是幾毫秒 (0-999)
getDay()	取得是星期幾 (0-6)
getTime()	取得從 1970-01-01 00:00:00 UTC 累計的毫秒數

JavaScript內建物件-Date物件

› 設定日期和時間相關的方法

方法	用途
setFullYear()	設定是幾年
setMonth()	設定是幾月 (0-11)
setDate()	設定是幾日 (1-31)
setHours()	設定是幾時 (0-23)
setMinutes()	設定是幾分 (0-59)
setSeconds()	設定是幾秒 (0-59)
setMilliseconds()	設定是幾毫秒 (0-999)
setTime()	用 timestamp milliseconds 設定是什麼日期時間

JavaScript內建物件-Array物件

- › Array物件提供了操作陣列的屬性與方法
 - 常用屬性有length，回傳陣列的長度
- › Array物件常用的方法有操作陣列、排序、新增刪除等

操作陣列方法	說明
includes(elm[, start])	傳回陣列是否包含參數elm，可搭配參數start來指定從哪個索引開始檢查
toString()	將陣列轉成字串物件
concat(arr)	傳回呼叫此方法陣列與參數arr合併的陣列
fill(value[, begin[, end]])	傳回以value填滿從索引begin~(end-1)的陣列
join([separator])	傳回陣列各個元素連接而成的字串，可以搭配separator參數來指定連接的字元
slice(begin[, end])	傳回索引為begin~(end-1)的元素組成的陣列

JavaScript內建物件-Array物件

› 排序

排序陣列方法	說明
sort()	傳回將陣列由小到大排序的結果
reverse()	傳回將陣列反序的結果

› 新增刪除

新增刪除陣列方法	說明
pop()	從陣列移除最後一個元素，並傳回該元素
shift()	從陣列移除第一個元素，並傳回該元素
push()	新增一個或多個元素到陣列的末端，並回傳陣列的新長度



Browser Object Model (BOM)瀏覽器物件模型

BOM物件-Window物件

- › Window物件提供可以存取和操作瀏覽器視窗的屬性和方法
- › 取得瀏覽器視窗的寬高屬性 (Window Size)

視窗屬性	說明
window.innerWidth	回傳視窗的可見寬度
window.innerHeight	回傳視窗的可見高度
window.outerWidth	回傳整個視窗的寬度(包含工具列、卷軸等)
window.outerHeight	回傳整個視窗的高度

BOM物件-Window物件

- › 瀏覽器開啟新視窗 `window.open(strUrl, strWindowName, [strWindowFeatures])`
 - 參數 `strUrl` 表示你要開啟的網址
 - 參數 `strWindowName` 是新視窗的名稱，可以用來設定給 `<a>` 和 `<form>` 標籤的 `target` 屬性。
 - 選擇性的參數 `strWindowFeatures` 用來設定新視窗的屬性 (attribute)，有下一頁列出的這些值可以設定
 - 將你想要的屬性指定為 `yes`，不想要屬性指定為 `no` 即可 (也可以用 1 或 0 表示)，不同的屬性用逗點，分隔開

BOM物件-Window物件

屬性	用途
width	視窗的寬度 (px)
height	視窗的高度 (px)
left	視窗位置距離螢幕左緣的距離 (px)
top	視窗位置距離螢幕上緣的距離 (px)
menubar	主選單是否出現
location	網址列是否出現
scrollbars	捲軸是否出現
status	狀態列是否出現
toolbar	工具列是否出現
titlebar	標題列是否出現
resizable	可否改變視窗的大小
fullscreen	是否打開成全螢幕模式

BOM物件-Window物件

- › 關閉瀏覽器視窗 `window.close()`
 - 這個方法只能用來關閉 `window.open()` 方法開啟的視窗
- › 移動瀏覽器視窗
 - `window.moveTo(x, y)`
 - › 用來移動瀏覽器視窗的位置到指定的(x, y)座標上
 - › 參數 x, y 的單位是 px，分別表示移動視窗位置至距離螢幕左緣和上緣多少的距離
 - `window.moveBy(deltaX, deltaY)`
 - › 將視窗做相對位置的移動
 - › 以視窗目前的位置為基準，分別要往左和往下移動多少距離 px
- › 基於瀏覽器安全限制，執行以上移動指令所在的網頁和開啟的新網頁，必須在同樣的網域下才可以操作

BOM物件-Location物件

- › location 物件提供存取瀏覽器頁面的網址 (URL) 相關資訊的屬性和方法
- › 可以透過這個物件取得或控制瀏覽器的網址、重新載入網頁、或導向其他網頁

BOM物件-Location物件

› 常用的屬性

屬性	說明
location.href	取得當前網頁的網址，還可以用來將網頁切換到新的網址
location.hostname	取得當前網頁的網域名稱 (domain name; host name)
location.pathname	取得當前網頁的網址路徑
location.search	取得當前網頁網址上的參數，如果沒有網址參數則會返回空字串。注意返回的參數包含問號？。
location.hash	取得當前網頁網址上的 hash 值，如果網址中沒有 hash 則會返回空字串。注意返回的 hash 包含井字號 #。
location.port	取得當前網址上的通訊埠值
Location.protocol	取得當前網址上的通訊協定

BOM物件-Location物件

› Location物件的方法

方法	說明
<code>location.reload(forcedReload)</code>	用來重新載入目前開啟的網頁。 參數 <code>forcedReload</code> 是一個布林值 (boolean)，如果是 <code>true</code> 用來強制瀏覽器從 <code>server</code> 取得最新的資料；預設是 <code>false</code> ，會優先從瀏覽器暫存檔 (cache) 中取得網頁資料。
<code>location.assign(url)</code>	跟 <code>location.href</code> 用途類似，可以在當前視窗載入一個新的網頁
<code>location.replace(url)</code>	用來在當前視窗載入一個新的網頁，但使用 <code>location.replace()</code> 來載入，當前網頁的瀏覽紀錄 (History) 會被新的網頁取代掉，也就是讓使用者沒辦法按「上一頁」回去。
<code>toString()</code>	將網址(<code>location.href</code> 屬性的值)轉換成字串

BOM物件-Navigator物件

- › Navigator 物件提供存取使用者的瀏覽器資訊的屬性
- › 這些屬性只能讀取無法寫入，常見屬性如下

屬性	說明
navigator.appName	用來取得瀏覽器的名稱，目前的瀏覽器 IE11+, Chrome, Firefox 和 Safari 都統一會返回 "Netscape"
navigator.appCodeName	用來取得瀏覽器的代碼名稱，例如Mozilla
navigator.appVersion	用來取得瀏覽器的版本號
navigator.cookieEnabled	用來檢查瀏覽器的 cookie 功能是否有開啟，返回布林值 true or false
navigator.onLine	用來檢查使用者的瀏覽器是否有連上網路或是斷線了，返回布林值 true or false
navigator.userAgent	用來取得瀏覽器完整的版本資訊
navigator.vendor	用來取得瀏覽器的廠商名稱，如果是 Chrome 瀏覽器會顯示 "Google Inc."

BOM物件-History物件

› History 物件提供存取瀏覽器的瀏覽歷程記錄的屬性和方法

屬性/方法	說明
history.length	取得使用者在當前視窗下，總共瀏覽了幾個網頁
history.back()	這個方法可以用來使瀏覽器回到上一頁
history.forward()	這個方法可以用來使瀏覽器回到下一頁
history.go(relativePosition)	這個方法可以用來明確指定瀏覽器要回去幾頁。 參數 relativePosition 是一個數字，表示相對於當前頁面，要往上幾頁 (負數) 或往下幾頁 (正數)。

BOM物件-Screen物件

- › Screen 物件提供存取使用者的螢幕畫面相關資訊的屬性和方法

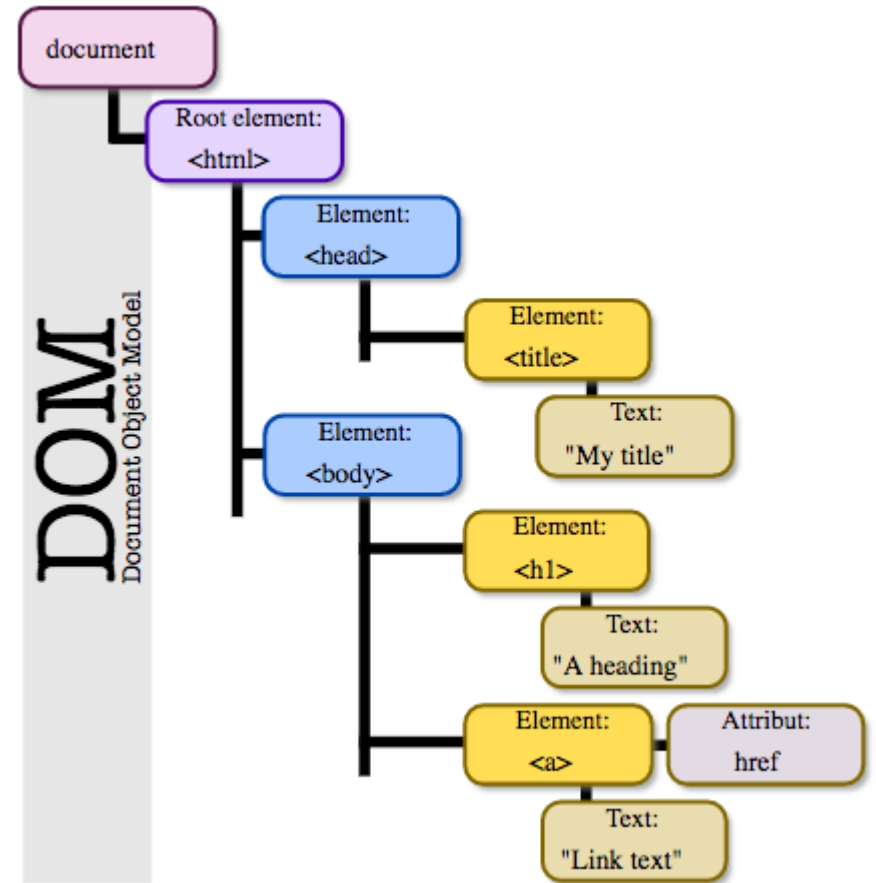
屬性	說明
screen.width	取得使用者螢幕的寬度，單位是 px
screen.height	取得使用者螢幕的高度，單位是 px
screen.availWidth	用來取得使用者瀏覽器可以使用的寬度，單位是 px
screen.availHeight	用來取得使用者瀏覽器可以使用的高度，單位是 px
screen.colorDepth	取得使用者螢幕的色彩深度/位數



Document Object Model (DOM) 文件物件模型

DOM文件物件模型

- › DOM (Document Object Model) 定義了一組標準 API讓我們可以用 JavaScript 對 HTML 文件做操作。
- › DOM 將一份 HTML 文件看作是一個**樹狀結構**的物件，讓我們可以方便直觀的存取樹中的**節點** (node) 來改變其結構、樣式 (CSS) 或內容等

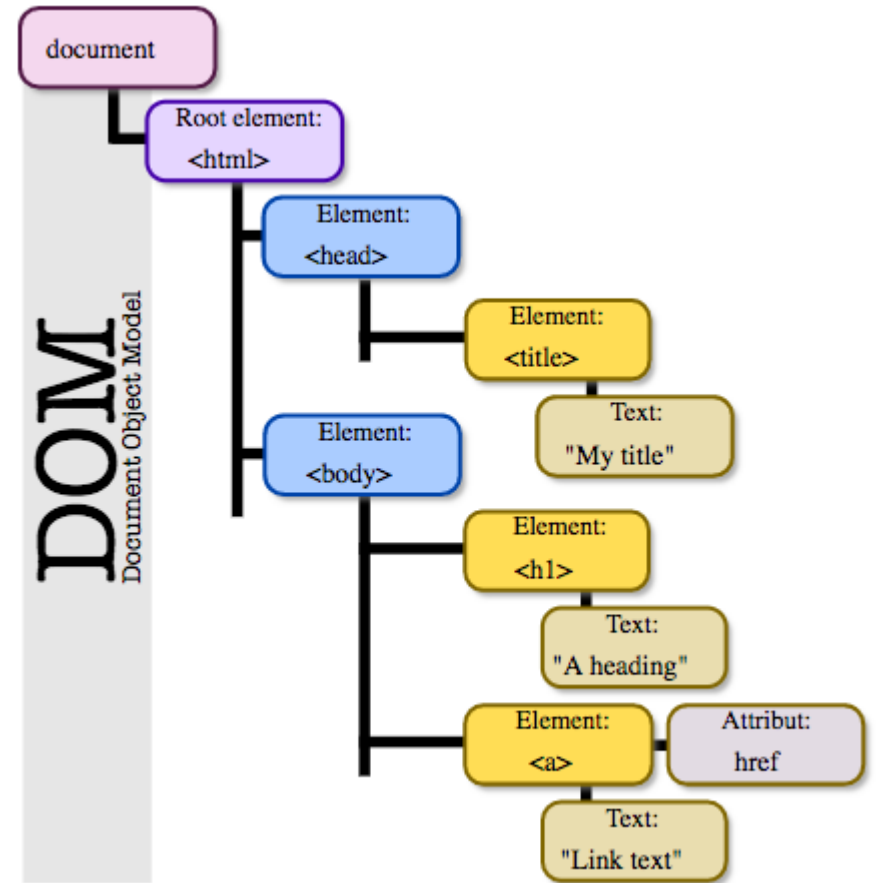


DOM文件物件模型

- › HTML DOM 規範中定義了這些類型的 API
 - 讓我們可以對 HTML 的元素 (element) 當作是 JavaScript 物件 (object) 來操作
 - 定義了 HTML 元素有哪些屬性 (properties) 可以來做存取
 - 定義了 HTML 元素有哪些方法 (methods) 可以來被操作
 - 定義了 HTML 元素事件 (events)，讓我們可以針對特定元素來綁定事件處理函式 (例如使用者按下滑鼠、在鍵盤打字都是所謂的事件)

DOM節點類型

- › document 物件是 DOM tree 的根節點，表示整份 HTML 文件
- › DOM 中的節點類型
 - 元素節點 (element nodes)
 - 文字節點 (text nodes)
 - 註解節點 (comment nodes)
 - 空白節點 (whitespace nodes，也是一種 text node)。
 - 屬性節點(attribute nodes)



DOM 查找元素

- › 通常你要存取 HTML 都是從 document 物件開始。
- › document.getElementById(id)
 - 用來根據 id 取得一個 HTML 元素。
- › document.getElementsByTagName(name)
 - 用來根據 HTML 標籤 (tag) 名稱取得所有這個標籤的元素集合
 - 返回的結果是一個像陣列 (array) 的物件
 - 每個 HTML DOM 元素也有 getElementsByTagName 方法，用來找該元素下面的子元素

DOM 查找元素

- › document.getElementsByName(name)
 - 用來取得特定名稱 (name) 的 HTML 元素集合
 - 返回的結果是一個像陣列 (array) 的物件
- › document.getElementsByClassName(names)
 - 用來取得特定類別名稱 (class name) 的 HTML 元素集合
 - 返回的結果是一個像陣列 (array) 的物件

DOM 查找元素

- › document.querySelector(selectors)
 - 用 CSS 選擇器 (CSS selectors) 來尋找符合條件且第一個找到的 HTML 元素
 - 每個 HTML DOM 元素也有 querySelector 方法，用來找該元素下面的子元素
- › document.querySelectorAll(selectors)
 - 可以用 CSS 選擇器 (CSS selectors) 來尋找所有符合條件的 HTML 元素集合
 - 返回的結果是一個像陣列 (array) 的物件
 - 每個 HTML DOM 元素也有 querySelectorAll 方法，用來找該元素下面的子元素

DOM tree 節點間位置的相互關係

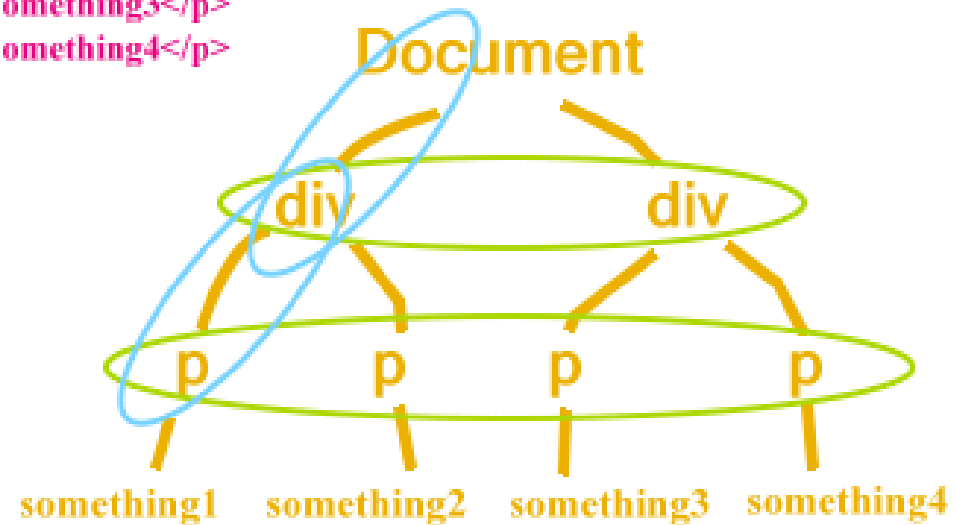
› DOM 元素節點之間的相互關係

- 父節點 (parent)
- 子節點 (child)
- 兄弟節點 (sibling)

```
<div>  
  <p>something1</p>  
  <p>something2</p>  
</div>  
<div>  
  <p>something3</p>  
  <p>something4</p>  
</div>
```

■ 父子關係 Parent & Childs (縱向)

■ 兄弟關係 Siblings (橫向)



DOM tree 子節點查找

› Node.childNodes

- 所有的 DOM 節點物件都有 childNodes 屬性，用來取得該元素下的所有子元素集合

› Node.children

- 差異在於 childNodes 返回的子元素會包含文字節點 (text nodes) 和註解節點 (comment nodes)，children 屬性則只會返回 HTML 元素節點

› Node.firstChild

- 來取得其下的第一個子節點或返回 null 表示沒有任何子節點。

› Node.lastChild

- 來取得其下的最後一個子節點或返回 null 表示沒有任何子節點。

DOM tree 父節點查找

- › Node.parentNode
 - 用來取得其父元素
 - 得到值可能會是一個
 - › 元素節點 (Element node)
 - › 根節點 (Document node)
 - › DocumentFragment 節點

DOM tree 兄弟節點查找

› Node.previousSibling

- 用來取得在目前節點前面的兄弟節點 (sibling)
- 返回 null 表示該節點已經是第一個子節點。

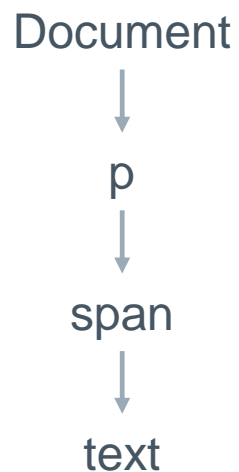
› Node.nextSibling

- 用來取得在目前節點後面的兄弟節點 (sibling)
- 返回 null 表示該節點已經是最後一個子節點。

DOM tree 附註

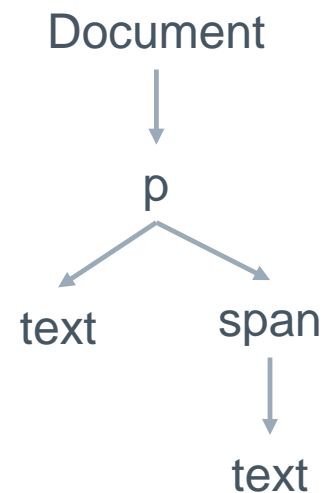
`<p id="foo">first span</p>`

› 沒有縮排的Tree



`<p id="foo">
 first span
</p>`

› 有縮排的Tree



DOM 節點物件的屬性

› Node.nodeType

- 可以取得該節點的類型，像是 elements (元素節點), text (文字節點) 或 comments (註解節點)

常數名稱	值	說明
Node.ELEMENT_NODE	1	表示 HTML 元素 (Element) 節點，像是 <p> 或 <div>
Node.TEXT_NODE	3	表示文字 (Text) 或屬性 (Attr) 節點
Node.COMMENT_NODE	8	表示註解節點 (Comment)
Node.DOCUMENT_NODE	9	表示根節點 (Document)
Node.DOCUMENT_TYPE_NODE	10	表示 DocumentType 節點，像是 <!DOCTYPE html>
Node.DOCUMENT_FRAGMENT_NODE	11	表示 DocumentFragment 節點

DOM 節點物件的屬性

- › Node.nodeName
 - 會返回節點的名稱

節點類型	nodeName 的值
Attr	值同 Attr.name
Comment	"#comment"
Document	"#document"
DocumentFragment	"#document-fragment"
DocumentType	值同 DocumentType.name，例如 "html"
Element	值同 Element.tagName
Text	"#text"

DOM 節點物件的屬性

› Element.tagName

- tagName 屬性只有 HTML 元素節點才有
- 在其他類型的節點 (例如 text nodes) 值則會是 undefined

› Node.nodeValue

- 元素節點沒有nodeValue
- 用來取得 text, comment 和 CDATA 節點的內容
- attribute 節點則會返回屬性內容

DOM 節點物件的屬性

› Element.innerHTML

- 用來取得一個 DOM 元素節點中的 HTML 內容
- innerHTML 屬性是可寫的，可以用來改變某元素下的 HTML 內容

› Element.outerHTML

- 用來取得一個 DOM 元素節點的 HTML 內容
- 跟 innerHTML 的差別在於 outerHTML 還會返回節點本身的 HTML 內容

› Element.innerText

- 和 innerHTML 用法類似，只是 innerText 會將 HTML 標籤去除

DOM 節點物件的屬性

› Node.textContent

- textContent 屬性和 innerText 用法類似，但有下面這些差異：
 - › textContent 會返回包含 `<script>` 和 `<style>` 的所有內容，但 innerText 不會。
 - › innerText 不會返回畫面中看不見的元素 (例如樣式是 `visibility: hidden;` 或 `display: none;`)，但 textContent 會返回。
- textContent 屬性是可寫的，可以用來改變某元素下的 HTML 內容

練習

- › 完成一個猜數字遊戲
 - guess_number.html
- › 讓使用者輸入一個1-30的整數
- › 用Javascript獲得使用者輸入的數字並判斷是否為正確答案
 - 正確答案就在結果顯示區域顯示猜對
 - 不是正確答案就在結果顯示區域顯示猜錯，並在提示區域顯示猜的數字比正確答案大或小