DOM-查找標籤

# 前言

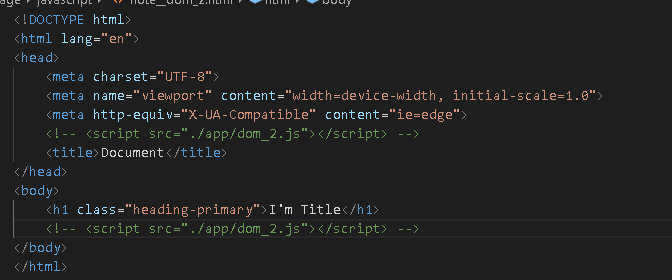
在上一則筆記裡面有稍微介紹過了DOM，這一篇就會來談談，具體上要怎麼使用DOM的API去找網頁內容(節點、html中的標籤)，還有怎麼去找自己的爸爸媽媽、兄弟姊妹。

## <script>要放哪裡?

其實這沒有標準答案，基本上愛放哪就放哪，不過一個不小心可能就會有問題囉!!可以放在<head>裏頭，也可以放在<body>裏頭，很常看到或是聽到的是

* 放在<head>裏頭
* 放在<body>裏頭

基本上這兩個位置都可以放，只不過差別在哪裡呢?我們拿個例子來說說，比如說今天在body裏頭有個<h1 class=”heading-primary”></h1>標籤.



我們在上一篇中得知可以使用DOM的API去操作網頁內容，首先我們可以使用「doucment.querySelector()」來只到我們需要的節點，要找到上面的h1標籤的話我們可以這樣做:

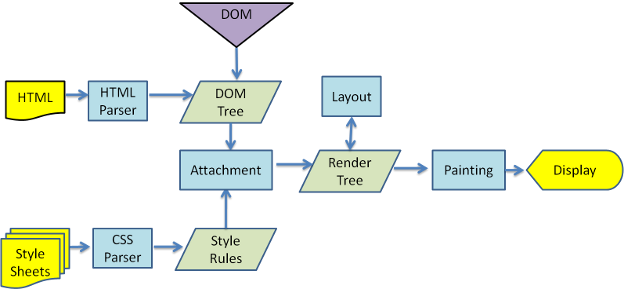
CODE:document.querySelector(‘.heading-primary’);

依照class來找到我們的h1標籤，然後我們可以透過textContent來去更改h1的內容，比如說我們把標籤改成HelloWord:

Code

再來我們把<script>放在body的最後面。耶!成功了，h1的內容成功的被改成「Hello!World!」，那我們放在<head>在試一次，結果什麼都沒有發生!Oh NO!怎麼會這樣，而且也沒有任何的報錯紀錄，真的是爛透了!!

冷靜，我們先來看看下面這張圖:



\

當一個網頁被載入到瀏覽器的時候，第一件事情做的就是去解析HTML文件，「由頭到尾」「由上而下」依序往下解讀，遇到<script>的時候會停止解析，開始執行<script>的內容，執行完畢後在繼續往下開始解析文件。

看完了上面這些後，知道為何我們剛剛把<script>放在<head>裏頭卻沒有發生任何事情的問題所在了嗎?在<head>裏頭的<script>嘗試去取得<h1>標籤，但是<h1>標籤還並沒有被解析出來，想當然而要去取得是取得不到的!!

那麼當我們放在<body>的最尾端的時候，此時<h1>標籤已經被解析出來了，所以就可以取得並且將內容更改成Hello!World!囉!!那麼是不是<script>就不適合放在<head>裏頭嗎?不是歐!當然會有其他方法可以去優化解決的，但今天的主題不再這裡，所以今後再說囉!!!

※會建議去了解網頁的整個渲染過程!可以看看這些網站(超連結)

* <https://www.itread01.com/content/1546714651.html>
* <http://taligarsiel.com/Projects/howbrowserswork1.htm>

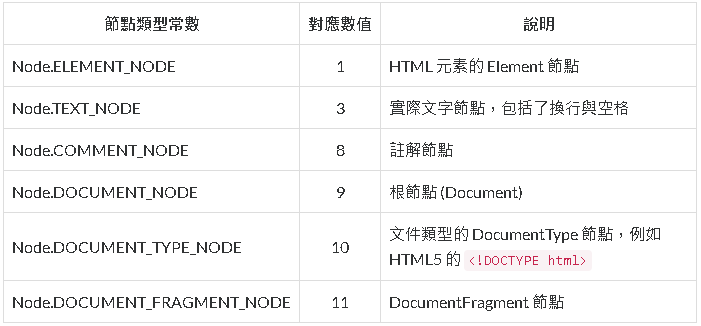
# 尋找節點

之前我們使用過了「doucument.querySelector」來查找節點，

除了這個方法之外，也有其他的方法可以查找噢!!

* document.getElementByID(‘xxx’);
* document.getElementByTagName(‘xxx’)
* document.getElementByClassName(‘xxx’);
* document.querySelector(‘xxx’);
* document.querySelectorAll(‘xxx’);

※在上一篇中有一張圖表，有節點類型常數與對應值，其中最常見下列幾種



可以透過類型常數或是對應值來判斷

# querySelector v.s getElementBy\*\*\*

剛剛我們說到了幾種方法，querySelector()或是getElementBy\*\*\*都是只會取得文件中的第一個(所以一定只會只有一個節點，所以沒有length、index屬性)。既然這兩個能夠達成相同的效果，那差在哪裡呢?!就差在getElementBy\*\*\*()並不是使用「CSS選擇器」的格式，比如說用ID查找節點

* getElementByID(‘id’)
* querySelector(‘#id’)

用ClassName查找

* getElementByClassName(‘classname’)
* querySelector(‘.classname’)

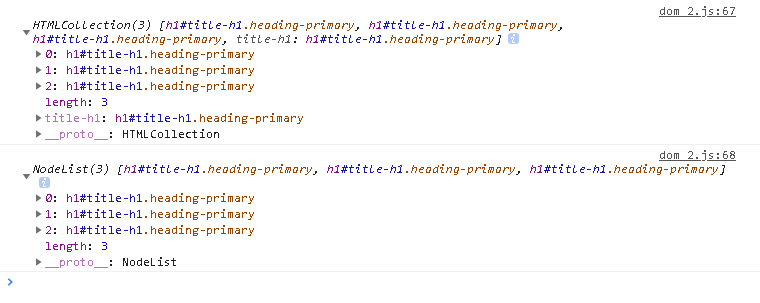
# getElementBy\*\*\*s v.s querySelectorAll

這兩個都可以找所有符合條件的節點，但是回傳的值要住一下，getElementBy\*\*\*s回傳的是「HTMLColleciton」;querySelectorAll回傳的是「Nodelist」，「HTMLColleciton」只收集Element節點，但是「Nodelist」除了Element節點之外，也含了文字節點、屬性節點等等。雖然不能使用陣列型別的method，但兩個都可以透過索引值來存取內容。

還有一點需要注意，HTMLCollenciton/NodeList大部分的情況下都是即時更新的，但是透過「querySelector」或是「querySelectorAll」取得的都是靜態的!

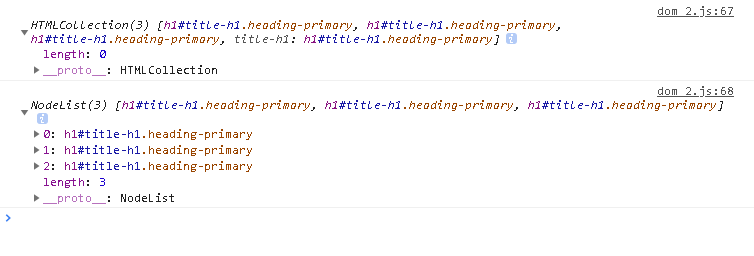
即時更新是什麼意思呢?就是會隨著DOM而改變，但是靜態的卻不會，舉個例子:

CODE



拿掉註解，清空body後

CODE



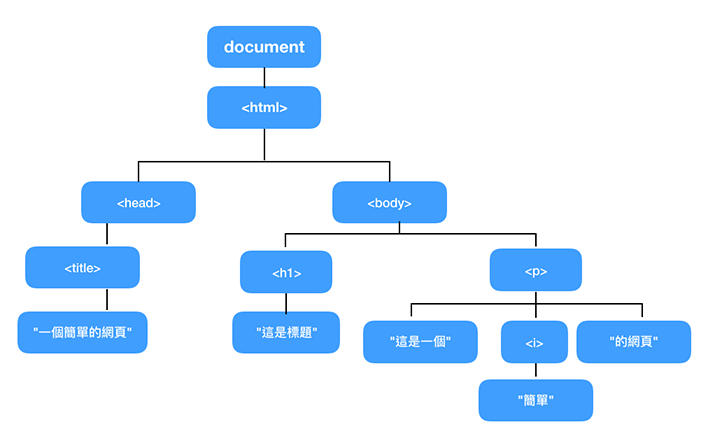
※「NodeList」物件是節點的集合，除了selectorAll之外，也可以藉由Node.childNodes屬性來取得，雖然有跟Array相似的特性，但它不是Array，所以也就不會有Array的method，如fifter、map等等。

# 遍歷(Travering)

除了直接透過ID或是CLASS來查找之外，我們也可以透過關係來找，上一篇有講到節點們主要有兩種關係

* 父子(parent and child)
* 兄弟 (sibling)

假設現在是這樣子的:



## 父子關係

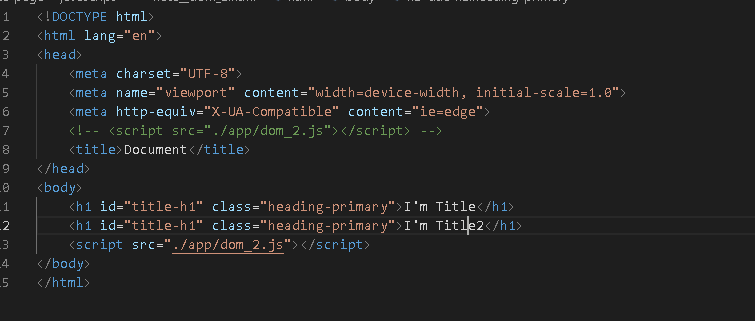
假如我現在是上圖中的<h1>，那麼我的父(Parent node，上層節點)就是<body>，而我的子(Child node，下層節點)就是textNode(“這是標題”)

## 兄弟關係

一樣的我是<h1>，那我的兄弟就是<p>啦，有同一個Parent node的同層節點，就是兄弟啦!!

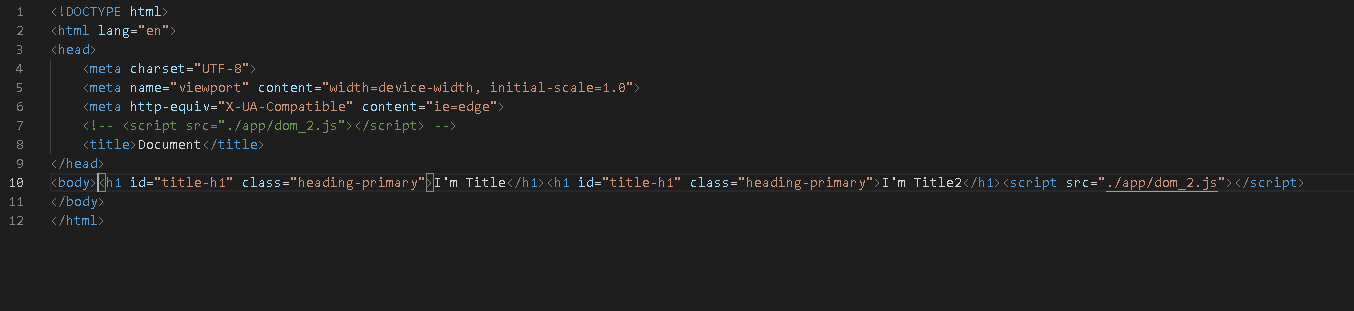
## Node.childNodes

所有的DOM節點物件都有這個屬性，而且此種屬性無法修改。我們可以透過Node.hasChildNodes()來檢查DOM是否有子節點:



CODE

結果出來後很有可能你會發現有textNode，只要把Html文件的換行拿掉變成:



這樣textNode就會不見囉!!

## parentNode/parentElement

我們可以這樣來找父節點，比如說:

CODE

※parentNode與parentElement的不同是，前者包括文字、註解節點等等;後者則是只有Element節點，而之後的找child or Sibiling都有這兩者之分。

## firstChild/firstElementChild

firstChild/firstElementchild可以找到第一個子節點，如果沒有子節點則回傳null。

CODE

## lastChild/lastElementChild

可以找第一個，也可以找最後一個。

CODE

## previousSibling/previousElementSibling

previousSibling可以找到自己的前一個節點

CODE

## nextSibling/nextElementSibling

找自己的下一個節點。

CODE

# End

這次的介紹就到這裡啦!下一期我們會介紹