

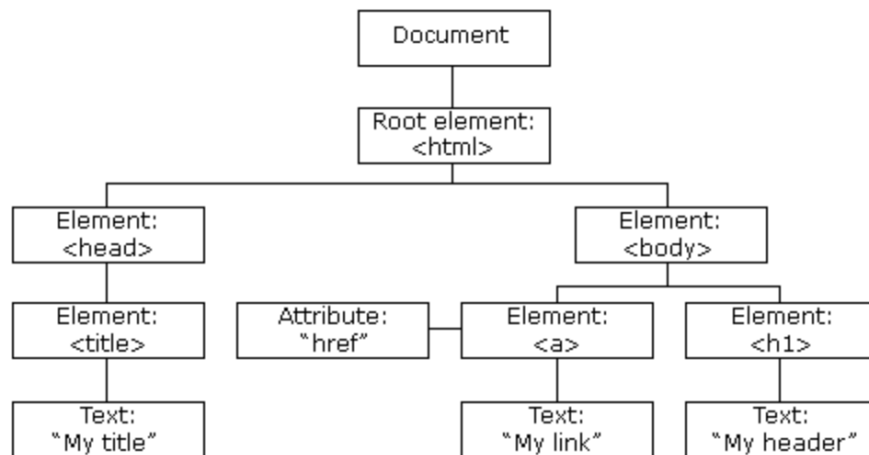


DOM與BOM

DOM

DOM : Document Object Model

- 樹狀結構



找第一個小孩

網頁結構如下

```
<body>
  <ul>
    <li>A</li>
    <li>B</li>
    <li>C</li>
    <li>D</li>
  </ul>
</body>
```

- A
- B
- C
- D

選到 `` 後找到第一個標籤小孩

```
document
.querySelector('ul')
.firstElementChild
.style
.backgroundColor = 'cyan'
```

有 first 就有 last

選到 `` 後找到第一個小孩（粉紅色區域）

```
document
.querySelector('ul')
.firstChild
```

這個位置是 `` 與第一個 `` 之間的換行與一堆的空白鍵

找第 N 個小孩

找第三個小孩（僅算標籤）

```
document
.querySelector('ul')
.children[2]
.style
.backgroundColor = 'cyan'
```

- A
- B
- C
- D

找第六個小孩（除標籤外，含文字內容）

```
document
.querySelector('ul')
.childNodes[5]
.style
.backgroundColor = 'cyan'
```

例如 `` 與下一行 `` 之間的換行與空白鍵

找父母

- A
- B
- C
- D

```
document
.querySelector('li')
.parentNode
.style
.backgroundColor = 'cyan'
```

```
document
.querySelector('li')
.parentElement
.style
.backgroundColor = 'cyan'
```

parentNode 與 parentElement 原則上一樣，只有在找 <html> 的父母時，
parentNode 傳回網頁 url，而 parentElement 傳回 null

找鄰居

同層的下一個標籤

```
document
.querySelector('li')
.nextElementSibling
.style
.backgroundColor = 'cyan'
```

同層的上一個標籤

```
document
.querySelector('li')
.parentElement
.lastElementChild
.previousElementSibling
.style
.backgroundColor = 'cyan'
```

注意 `nextSibling` 與 `previousSibling` 會連
標籤與標籤間的換行與空白一起找

- A
- B
- C
- D

- A
- B
- C
- D

```
<body>
  <ul>
    <li>A</li>
    <li>B</li>
    <li>C</li>
    <li>D</li>
  </ul>
</body>
```

使用迴圈找小孩

將 `` 的最後兩個選項改背景顏色

```
const children = document.querySelector('ul').children
for (i = children.length - 1; i > children.length - 3; i--) {
  children[i].style.backgroundColor = 'cyan'
}
```

偶數列改顏色

```
const children = document.querySelector('ul').children
for (i = 0; i < children.length; i+=2) {
  children[i + 1].style.backgroundColor = 'cyan'
}
```

- A
- B
- C
- D

- A
- B
- C
- D

修改內容

將選項 A 的內容改為 AA

```
document
.querySelector('li')
.firstChild
.textContent = 'AA'
```

加不加都一樣

- AA
- B
- C
- D

在 前面加上標題

```
document
.body
.firstChild
.textContent = '選項'
```

一定要加

選項

- A
- B
- C
- D



附加一個小孩

在父元件的最後加上一個 child

```
const new_element = document.createElement('li')
new_element.textContent = "NEW"
new_element.style.color = 'red'

document
  .querySelector('ul')
  .append(new_element)
```

- A
- B
- C
- D
- **NEW**

插入新元素

在選項 D 前面插入新元素

```
const new_element = document.createElement('li')
new_element.textContent = "NEW"
new_element.style.color = 'red'
```

```
const itemD = document.querySelector('ul').children[3]
```

```
itemD
.parentNode
.insertBefore(
  new_element, itemD
)
```

只有 insertBefore() 函數

- A
- B
- C
- **NEW**
- D

練習

將 `` 標籤前方的「選項」改為使用 `<h1>` 標籤樣式

選項

- A
- B
- C
- D

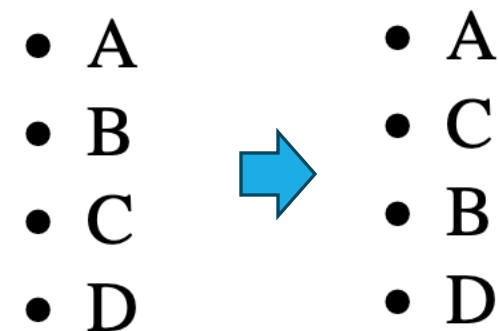
搬移元素

只要元件存在，使用 `appendChild()` 或是 `insertBefore()` 時就變成搬移

例如：將選項 B 與選項 C 對調（相當於將 B 插入到 D 的前面）

```
const itemB = document.querySelectorAll('li')[1]
const itemD = document.querySelectorAll('li')[3]
```

```
itemD
.parentNode
.insertBefore(itemB, itemD)
```



刪除

將選項 B 刪除

```
const itemB = document.querySelectorAll('li')[1]  
itemB.remove()
```

刪除 中最後兩個選項

```
const items = document.querySelectorAll('ul li')  
items[items.length - 1].remove()  
items[items.length - 2].remove()
```

TABLE 有專屬的函數

表格架構複雜，有專屬的函數操作 DOM，例如 `insertRow()`、`insertCell()`

```
const data = ['A', 'B', 'C', 'D', 'E', 'F']
const table = document.createElement("table")
table.setAttribute('border', '1')
document.body.appendChild(table)
```

```
for (row = 0; row < 2; row++) {
  const r = table.insertRow()
  for (col = 0; col < 3; col++) {
    const cell = r.insertCell()
    cell.textContent = data[row * 3 + col]
  }
}
```

A	B	C
D	E	F

練習

購物車頁面操作

按 inc 按鈕數量 +1，按 dec 按鈕數量 -1，若數量為 0 時該項目刪除

項次	名稱	數量	操作
1	鉛筆	7	inc dec
2	原子筆	4	inc dec



BOM

BOM - BROWSER OBJECT MODEL

Window：瀏覽器本身的視窗

Screen：取得螢幕資訊，例如解析度或是可顯示的顏色數量

Location：網址列

History：歷史瀏覽資訊、控制上一頁、下一頁

Navigator：取得正在瀏覽的那個頁籤有關資訊

Popup Alert：alert、prompt、confirm

Timing：setTimeout、setInterval

Cookie：本地端資料儲存，最多 4K，通常由後端存取

WINDOW

常用方法

開新頁籤 `open()`

有些網站會拒絕關閉，
例如 Google

```
w = window.open('https://www.apple.com', '_blank')
```

關閉頁籤 `close()`

```
w.close()
```

LOCATION

透過程式碼切換到另外一個網頁

```
location.href = 'https://www.apple.com'
```