

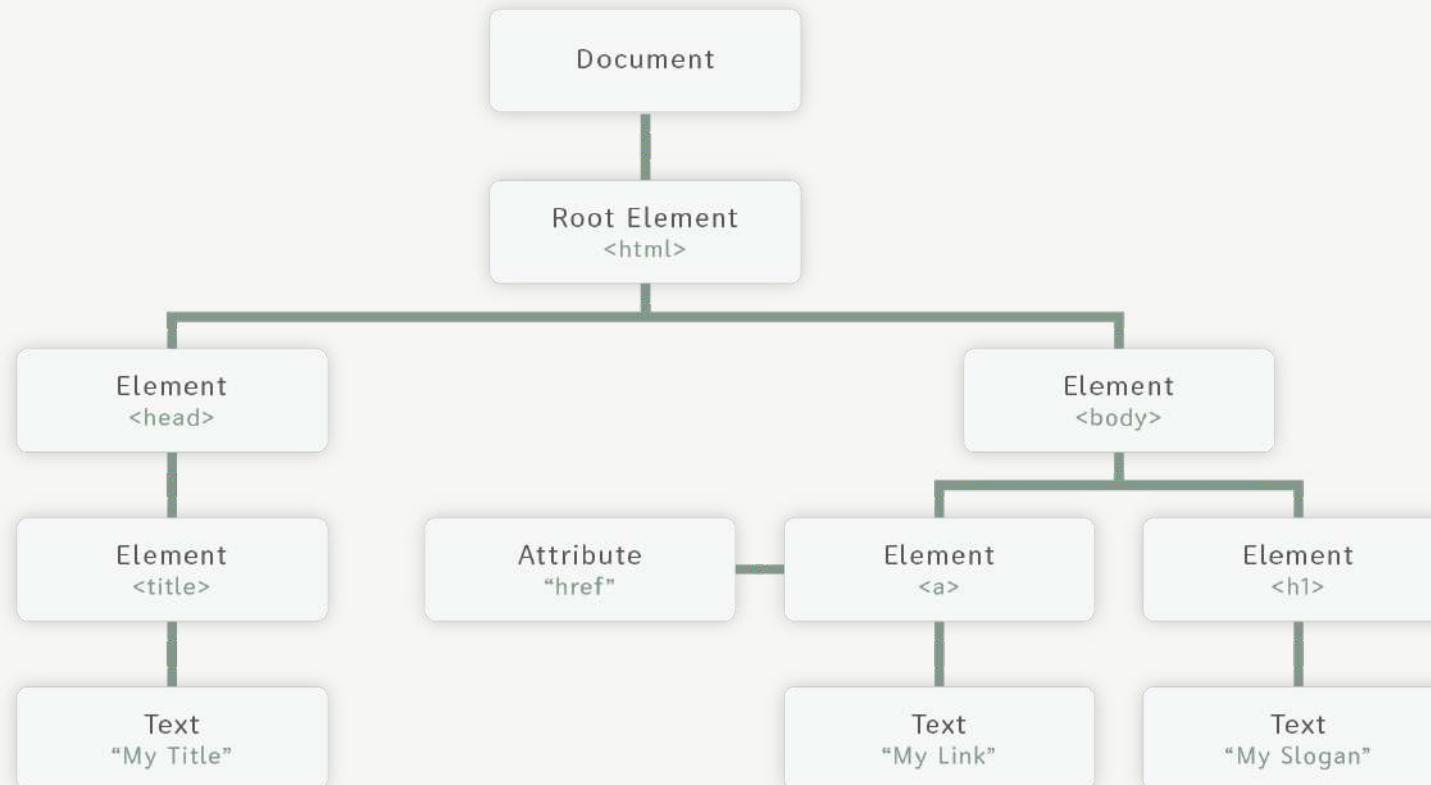
# 使用 JS 操作 DOM

從JavaScript取得或改變網頁元素

# 什麼是 DOM ?

全名 Document Object Model 中文為文件物件模型。

簡單來說就是把 Document (也就是 HTML) 轉換成 Object (物件)，是瀏覽器提供使用 JavaScript 時可以操作 Document 上元素的橋樑。



# 取得 DOM 元素

## .getElementsByName()

說明

取得所有指定屬性的元素

實際用例

```
1. // 取得 HTML 中的所有 'div' 標籤  
document.getElementsByTagName('div')  
  
2. // 取得 HTML 中的第三個 'div' 標籤  
document.getElementsByTagName('div')[2]
```

# 取得 DOM 元素

`.getElementById( )`

## 說明

取得指定的 id 元素

## 實際用例

```
1. // 取得 HTML 中 id 為 'navbar' 的元素  
document.getElementById('navbar')
```

`.getElementsByName( )`

## 說明

取得所有包含指定 class 的元素

## 實際用例

1. // 取得 HTML 中所有 class 包含 'block' 的元素  
`document.getElementsByClassName('block')`
2. // 取得 HTML 中第一個 class 包含 'block' 的元素  
`document.getElementsByClassName('block')[0]`
3. // 取得 HTML 中所有 class 包含 'block1 block2' 的元素  
`document.getElementsByClassName('block1 block2')`

# 取得 DOM 元素

## .querySelector( )

### 說明

取得第一個指定選擇器的元素

### 實際用例

```
1. // 取得第一個選擇器 '#navbar a' 匹配的元素  
document.querySelector('#navbar a')  
  
2. // 取得第一個選擇器 '.block > .content' 匹配的元素  
document.querySelector('.block > content')
```

## .querySelectorAll( )

### 說明

取得所有指定選擇器的元素

### 實際用例

```
1. // 取得所有選擇器 '.block p' 匹配的元素  
document.querySelectorAll('.block p')  
  
2. // 取得第三個選擇器 '.block1, .block2' 匹配的元素  
document.querySelectorAll('.block1, .block2')[2]
```

# 改變元素 CSS 樣式 (不推薦)

## .style

說明

直接更改元素css樣式

實際用例

```
1. // 方法一  
HTMLElement.style['padding'] = '20px'  
2. // 方法二  
HTMLElement.style.paddingTop = '20px'
```

# 改變元素 Class ( 較推薦 )

`.classList.add( )`

說明

新增指定的 class

實際用例

```
1. // 於指定元素新增class 'active'  
HTMLElement.classList.add('active')  
2. // 於指定元素新增多個 class  
HTMLElement.classList.add('active', 'move')
```

`.classList.remove( )`

說明

刪除指定的 class

實際用例

```
1. // 於指定元素刪除 class 'active'  
HTMLElement.classList.remove('active')
```

# 改變元素 Class

`.classList.toggle( )`

## 說明

判斷有無指定的 class，新增或刪除該 class

## 實際用例

```
1. // 判斷指定元素有無 class 'active'，新增或刪除 'active'  
HTMLElement.classList.toggle('active')
```

`.classList.contains( )`

## 說明

判斷是否包含指定的 class

## 實際用例

```
1. // 判斷指定元素有無 class 'active'，回傳true或false  
HTMLElement.classList.contains('active')
```

# 改變元素內容

## .textContent

### 說明

取得或以純文字重新設置指定元素內文字

### 實際用例

```
1. // 取得為傳值為 'Hello World!!!'  
  
HTMLElement = '<h1>Hello World<span  
style="display: none;">!!!</span></h1>'  
console.log(HTMLElement.textContent)  
  
2. // 將指定元素內 HTML 取代為 'Hello'  
  
HTMLElement.textContent = 'Hello'
```

## .innerText

### 說明

取得或以純文字重新設置指定元素內文字（受CSS影響）

### 實際用例

```
1. // 取得為傳值為 'Hello World'  
  
HTMLElement = '<h1>Hello World<span  
style="display: none;">!!!</span></h1>'  
console.log(HTMLElement.innerText)  
  
2. // 將指定元素內 HTML 取代為 'Hello'  
  
HTMLElement.innerText = 'Hello'
```

# 改變元素內容

## .innerHTML

### 說明

取得或重新設置指定元素內的 HTML 標籤

### 實際用例

```
1. // 將指定元素內 HTML 取代為 '<h2>Hello</h2>'  
HTMLElement.innerHTML = '<h2>Hello</h2>'
```

## .outerHTML

### 說明

取得或重新設置包含指定元素的 HTML 標籤

### 實際用例

```
1. // 將指定元素取代為 '<h2>Hello</h2>'  
HTMLElement.outerHTML = '<h2>Hello</h2>'
```

# 插入元素（較新）

## .append( 元素 )

### 說明

於指定元素內的最後方插入元素

### 實際用例

```
1. // 在指定元素內的最後方插入 '<span>Hello</span>'  
const el = document.createElement('span')  
el.innerHTML = 'Hello'  
Element.append(el)
```

## .prepend( 元素 )

### 說明

於指定元素內的最前方插入元素

### 實際用例

```
1. // 在指定元素前方插入 '<span>Hello</span>'  
const el = document.createElement('span')  
el.innerHTML = 'Hello'  
Element.prepend(el)
```

# 插入元素

`.appendChild( 元素 )`

說明

於指定元素內的最後方插入元素

實際用例

```
1. // 在指定元素內的最後方插入 '<span>Hello</span>'  
const el = document.createElement('span')  
el.innerHTML = 'Hello'  
Element.appendChild(el)
```

`.insertBefore( 元素, 指定元素 )`

說明

將新增的元素插入指定元素前方

實際用例

```
1. // 在指定元素前方插入 '<span>Hello</span>'  
const el = document.createElement('span')  
el.innerHTML = 'Hello'  
Element.insertBefore(el, HTMLElement)
```

# 刪除元素

.removeChild( 元素 )

說明

刪除元素內的指定子元素

實際用例

```
1. // 刪除指定元素內的 'a' 標籤  
Element.removeChild(document.querySelector('a'))
```

# 其他選取元素的方法

**.children**

說明

取得指定元素的所有子元素

**.firstChild**

說明

取得指定元素的第一個子元素  
受空格符影響

**.lastChild**

說明

取得指定元素的最後一個子元素  
受空格符影響

**.parentNode**

說明

取得指定元素的父元素

**.closest( )**

說明

取得最近的指定父元素