var ***john*** = {  
 firstName: 'John',  
 lastName: 'Doe'  
};  
  
for (var ***k*** in ***john***) {  
 ***console***.log(***k*** + ':' + ***john***[***k***]);  
}

結果 :

firstName: John  
lastName: Doe

**border-collapse 是否合併表格邊框**

//

element**.onclick = function**

element**.onmouseover = function**

element**.onmouseout = function**

// 獲取屬性

Element.屬性

Element**.getAttribute(‘**屬性**’)**

// 設定屬性值

element**.setAttribute(‘**屬性**’ , ’**值**’)**

// 獲取自定義屬性

element**.getAttribute(‘data-**index**’)**

element**.dataset.**index

// 父子節點

node**.parentNode**

node**.childNode**

// 選擇節點

parentNode**.firstChild**

parentNode**.lastChild**

parentNode**.firstElementChild**

parentNode**.lastElementChild**

// 兄弟節點

node**.nextSibling**

node**.previousSibling**

node**.nextElementSibling**

node**.previousElementSibling**

// 新增節點

document**.createElement('**tagName**')**

// 附值節點

node**.appendChild(child)**

node**.inserBefore(**child**,** 指定元素**)**

// 移除節點

node**.removechild(**child**)**

// 複製節點

node**.cloneNode(ture)**

// 創建元素

Documenet**.write()** 頁面重繪

// 方法監聽註冊

eventTarget**.addEventListener(‘**type**’,** listener{函數}, 事件流) // 可重疊執行

// 刪除註冊事件

eventTarget**.onclick = null** // 傳統作法

eventTarget**.removeEventListener(‘**type**’,** listener{函數}, 事件流)

// 事件流 (事件執行順序)

冒泡順序 事件流: false, 事件從內到外 son -> father -> div -> body -> html -> document

捕獲順序 事件流: true, 事件從外到內 document -> html -> body -> div -> father -> son

//

eventTarget**.addEventListener(‘**click**’,** function(e){

console.log(this); // 返回綁定的事件

console.log(e**.target)**; // 返回點擊的事件

})

// 阻止默認動作

e**.preventDefault()**

// 阻止冒泡

e**.stopPropagation()**