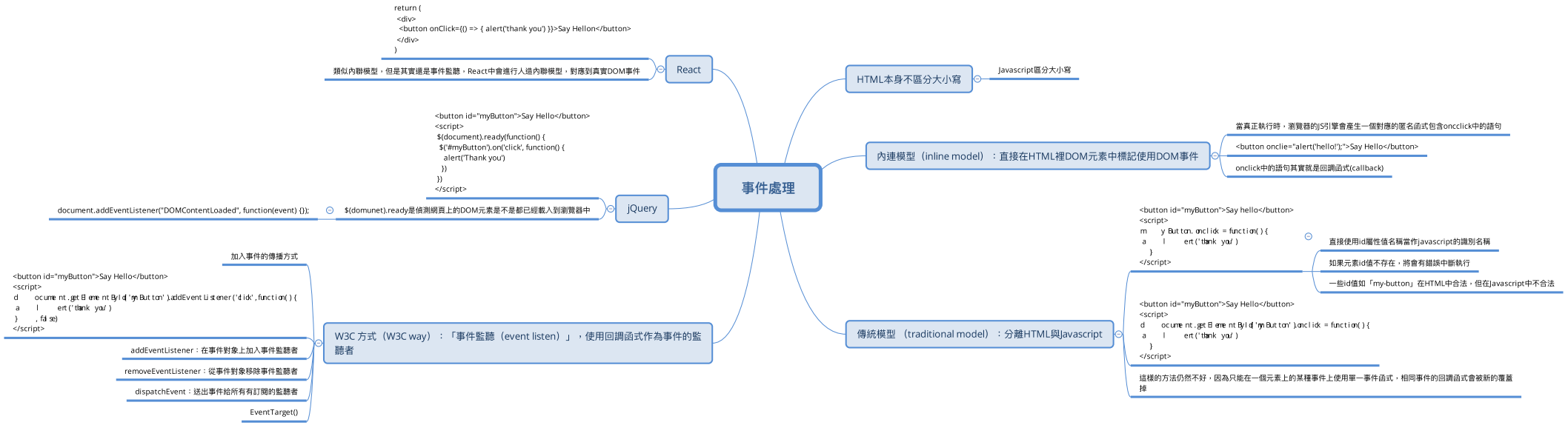
事件處理
React
return (

<button onClick={() => { alert('thank you') }}>Say Hellon</button>

)
類似內聯模型，但是其實還是事件監聽，React中會進行人造內聯模型，對應到真實DOM事件
jQuery
<button id="myButton">Say Hello</button>
<script>
$(document).ready(function() {
$('#myButton').on('click', function() {
alert('Thank you')
})
})
</script>
$(domunet).ready是偵測網頁上的DOM元素是不是都已經載入到瀏覽器中
document.addEventListener("DOMContentLoaded", function(event) {});
W3C 方式（W3C way）：「事件監聽（event listen）」，使用回調函式作為事件的監聽者
加入事件的傳播方式
<button id="myButton">Say Hello</button>
<script>
d ocume nt . get El eme nt By Id('myBu tton').addEvent Li stener ('c lick' , function( ) {
a l ert ('thank you')
} , fa lse)
</script>
addEventListener：在事件對象上加入事件監聽者
removeEventListener：從事件對象移除事件監聽者
dispatchEvent：送出事件給所有有訂閱的監聽者
EventTarget()
傳統模型（traditional model）：分離HTML與Javascript
<button id="myButton">Say hello</button>
<script>
m y Button. onclick = function( ) {
a l ert ('thank you')
}
</script>
直接使用id屬性值名稱當作Javascript的識別名稱
如果元素id值不存在，將會有錯誤中斷執行
一些id值如「my-button」在HTML中合法，但在Javascript中不合法
<button id="myButton">Say Hello</button>
<script>
d ocume nt . get El eme nt By Id('myBu tton').onclick = function( ) {
a l ert ('thank you')
}
</script>
這樣的方法仍然不好，因為只能在一個元素上的某種事件上使用單一事件函式，相同事件的回調函式會被新的覆蓋掉
內連模型（inline model）：直接在HTML裡DOM元素中標記使用DOM事件
當真正執行時，瀏覽器的JS引擎會產生一個對應的匿名函式包含oncclick中的語句
<button onclie="alert('hello!');">Say Hello</button>
onclick中的語句其實就是回調函式(callback)
HTML本身不區分大小寫
Javascript區分大小寫