

JSX 語法

JS 是 JavaScript，X 是 XML
一種針對 JavaScript 語言語法的擴充

- > React自創的在JS程式碼中建立DOM元素與使用自訂元件標記的語法
- > JSX中使用花括號({})嵌入JS表達式，具有求值運算作用
- > 是React.createElement方法的簡寫法，只能在React程式碼裡撰寫
- > 需要透過babel編譯過才能在JS引擎中執行



```
const element = <h1>Hello React!</h1>
```



BABEL 編譯



```
'use strict'  
  
const element = React.createElement('h1', null, 'Hello React!')
```

JSX 語法

一個render方法中 (函式型元件指最後 return 部份) 只能有一個根元素(父母元素)回傳

錯誤語法

```
return (  
  <h1>Title One</h1>  
  <p>Some Text</p>  
)
```

正確語法

```
return (  
  <>  
    <h1>Title One</h1>  
    <p>Some Text</p>  
  </>  
)
```

這與babel編譯為createElement方法有關，將多個DOM元素包裹在一個最外層的元素即可解決

JSX 語法

React.Fragment 元件(<>...</>)
不會產生多餘的render標記



```
return (  
  <div>  
    <h1>Title One</h1>  
    <p>Some Text</p>  
  </div>  
)
```



```
return (  
  <>  
    <h1>Title One</h1>  
    <p>Some Text</p>  
  </>  
)
```

當元件很多層級與分散很多時，會產生多餘的div標記，有可能會造成css樣式問題或效能的影響

JSX 語法

自我封閉 HTML 元素標記一定需要最後結尾
封閉標籤(`/>`)

一定要加上這個標記，
不然會有錯誤發生



```
const imgTag =   
const inputTag = <input type="text" name="title" />  
const brTag = <br />  
const hrTag = <hr />
```

JSX 語法

英文大寫開頭的標記為自訂元件標記
英文小寫開頭的標記為HTML使用的標記



```
{/* 內建HTML元素 */}
```

```
<h1>Title One</h1>
```

```
<p>Some Text</p>
```

```
{/* 自訂元件 */}
```

```
<MyButton title="One" />
```

```
<MyComponent />
```

JSX 語法

元件的加入事件為開頭為 on 的屬性
值必定為一個函式



```
<h2 onClick={() => setTotal(total + 1)}>{total}</h2>
```

JSX 語法

註解方式

VSCode中選定後按下「Ctrl + /」
(macOS為「CMD + /」)



```
return (  
  <>  
    { /* 這是一個註解 */ }  
    { /* <p>Some Text</p> */ }  
  </>  
)
```

JSX 語法

JS關鍵字的屬性需要改名稱，最常見的是
class -> className 與 for -> htmlFor

寫為class還是能正常執行，
但在執行時主控台會出現警告



```
<h1 class="title1">Title</h1>  
<label htmlFor="namedInput">Name:</label>  
<input id="namedInput" type="text" name="name" />
```


JSX 語法

Style 屬性為物件值

建議少用內聯 style 屬性，改為 CSS 類別指定



```
<p style={{color:'red', fontSize: '10px', backgroundColor:'#ffffff'}}>Title</p>
```

JSX 語法


使用花括號({})來嵌入JS變數或表達式



```
<h1>{1 + foo}</h1>
<MyButton
  text={'Click Me!'}
  onClick={() => alert('Hello')}
/>
```

JSX 語法

在JSX語法中要嵌入陣列
經常使用 map 方法



```
<>
  {students.map((student, index) => (
    <p key={index}>Hello, {student.name}!</p>
  ))}
</>
```

JSX 語法

列表項目標記，要使用key屬性



```
const todoItems = todos.map(  
  (todo) => <li key={todo.id}>{todo.text}</li>  
)
```

Key 與React要進行列表項目標記處理的最佳化有關，通常會出現在使用map方法與””、“<td>”等

JSX 語法

if 流程控制的表達式語法
(使用邏輯與 &&)



```
const count = 0
```

```
return <div>{count && <h1>Messages: {count}</h1>}</div>
```

意思為「如果 count 為 true，則呈現後面接著的元素」
(使用 Truthy 與 Falsy 來判斷)

JSX 語法

if...else 流程控制的表達式語法
(使用三元表達式`?:`)



```
<>
  {isLoggedIn ? (
    <LogoutButton onClick={this.handleLogoutClick} />
  ) : (
    <LoginButton onClick={this.handleLoginClick} />
  )}
</>
```