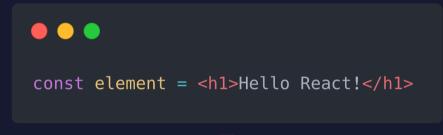
JS 是 JavaScript,X 是 XML 一種針對 JavaScript 語言語法的擴充

- 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





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

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>
    Some Text
)
```

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

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

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

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

自我封閉 HTML 元素標記一定需要最後結尾 封閉標籤(`/>`)

```
const imgTag = <img src="sample.jpg" />
const inputTag = <input type="text" name="title" />
const brTag = <br />
const hrTag = <hr />
```

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>
Some Text
{/* 自訂元件 */}
<MyButton title="One" />
<MyComponent />
```

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>

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

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。 但在執行時主控合會出現警告

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

Style屬性為物件值 建議少用內聯 style 屬性,改為 CSS 類別指定



Title

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')}
/>
```

在JSX語法中要嵌入陣列 經常使用 map 方法

列表項目標記,要使用key屬性

```
const todoItems = todos.map(
  (todo) => {todo.text}
)
```

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""、""等

if 流程控制的表達式語法 (使用邏輯與 &&)



if...else 流程控制的表達式語法 (使用三元表達式`?:`)