# State(狀態)

是會因為使用者操作後,而不斷改變的資料值



# State(狀態)

與使用者介面(UI)的互動有關 設計元件前應**以狀態為最優先的思考重點** 

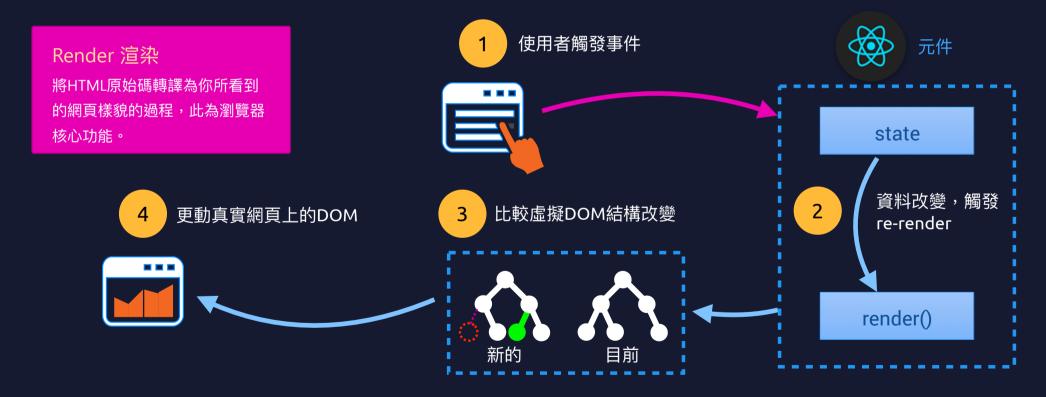
- 狀態為元件內部私有資料,其它元件無法 直接存取到
- > 類別型元件的狀態需定義在建構式之中
- 函式型元件的狀態需定義在函式主體區塊 最上方,使用"解構指定值"語法

```
constructor() {
   super()
   this.state = {
     total: 0,
   }
}

const [total, setTotal] = useState(0)
```

# State(狀態)

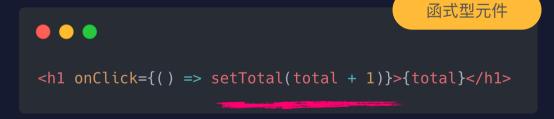
元件內部的私有資料,只有透過改變狀態 (setState) 才能改變真實網頁上的DOM



#### setState

唯一能更動 State(狀態) 的方法

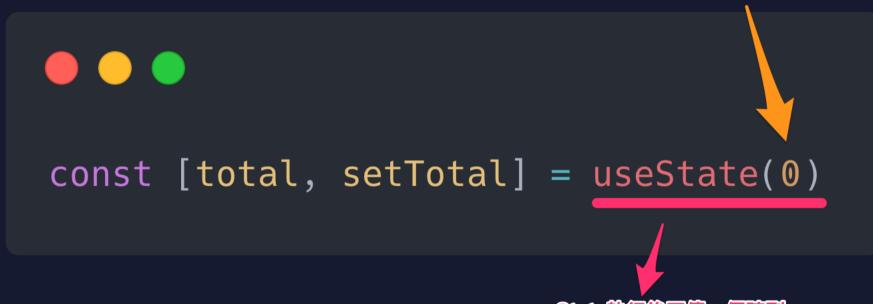
- 不可直接使用JS中的指定值運算語法來更動(ex. x = 1)
- 更動網頁上的UI -> State(狀態) 需要被改 變 -> 要呼叫 setState 方法
- 類別型元件與函式型元件語法有不少差 異,使用時務必注意(ex. 傳入參數值不同)
- > setState方法執行是有異步(非同步)特性, 使用時要注意程式執行時的邏輯



## useState 函式型元件勾子(hooks)

使用ES6中解構賦值的新語法

初始化值設定在傳入參數



useState執行後回傳一個陣列 [獲得值的變數。設定值的函式]

## setState 類別型元件

狀態本身就是物件值, setState也用物件值設定

```
phone: '0911-222-333',
         <h1>類別型元件</h1>
           value={this.state.name}
           onChange={(event) => this.setState({ name: event.target.value })}
           value={this.state.phone}
          onChange={(event) => this.setState({ phone: event.target.value })}
```

### setXXX 函式型元件

狀態不一定是什麼資料類型,注意使用物件值 時會被°覆蓋°(對照類別型元件為°合併°)