

# 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)

行進速度  
輪胎磨損情況  
行進距離  
...



## Props(屬性)

車身尺寸  
品牌  
顏色  
重量  
...

## State(狀態)

目前輸入的文字  
是否正在輸入文字中  
...

文字輸入框

## Props(屬性)

寬度  
高度  
預設文字  
...

# 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

## 類別型元件

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

## 函式型元件

```
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

## Render 渲染

將HTML原始碼轉譯為你所看到的  
網頁樣貌的過程，此為瀏覽器  
核心功能。



# 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

類別型元件

```
<h1 onClick={() => this.setState({ total: this.state.total + 1 })}>
  {this.state.total}
</h1>
```

函式型元件

```
<h1 onClick={() => setTotal(total + 1)}>{total}</h1>
```

# 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



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

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

```
1 constructor() {  
2   super()  
3   this.state = {  
4     name: 'Eddy',  
5     phone: '0911-222-333',  
6   }  
7 }  
8  
9 render() {  
10  return (  
11    <>  
12      <h1>類別型元件</h1>  
13      <input  
14        type="text"  
15        value={this.state.name}  
16        onChange={(event) => this.setState({ name: event.target.value })}  
17      />  
18      <input  
19        type="text"  
20        value={this.state.phone}  
21        onChange={(event) => this.setState({ phone: event.target.value })}  
22      />  
23    </>  
24  )  
25 }
```

# 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’)

```
1 const [data, setData] = useState({ name: 'Eddy', phone: '0911-222-333' })
2
3 return (
4   <>
5     <h1>函式型元件</h1>
6     <input
7       type="text"
8       value={data.name}
9       onChange={(event) => setData({ ...data, name: event.target.value })}
10    />
11    <input
12      type="text"
13      value={data.phone}
14      onChange={(event) => setData({ ...data, phone: event.target.value })}
15    />
16  </>
17 )
```

需要自行合併物件值，不然其它  
物件屬性會被丟棄