

# 可控(controlled) vs 不可控(uncontrolled)

	可控(controlled)	不可控(uncontrolled)
說明	由 React 控管資料狀態的表單元素	脫離 React 掌控狀態(原本的DOM)的表單元素，需由開發者自行控管資料的表單元件
應用搭配	State(狀態) / 改變State的事件	Refs(參照)
資料動態或立即檢查	容易達成	送出時才檢查/開發者自行處理
資料管理	由父母元件管理	DOM中由開發者自行管理
建議使用	大部份情況	少部份情況 (ex.有需要整合非react應用時)

# 常用表單元素

種類	標記	值屬性	事件callback	更新事件值 (針對 callback)	備註
文字輸入框	<code>&lt;input type="text" /&gt;</code>	<code>value="string"</code>	<code>onChange</code>	<code>e.target.value</code>	
複選框 (核取方塊)	<code>&lt;input type="checkbox" /&gt;</code>	<code>checked={boolean}</code>	<code>onChange</code>	<code>e.target.checked</code>	
選項按鈕	<code>&lt;input type="radio" /&gt;</code>	<code>checked={boolean}</code>	<code>onChange</code>	<code>e.target.checked</code>	
文字輸入區域	<code>&lt;textarea /&gt;</code>	<code>value="string"</code>	<code>onChange</code>	<code>e.target.value</code>	與HTML不同，用value代表其中輸入文字
下拉式選單	<code>&lt;select&gt;</code>	<code>value="option value"</code>	<code>onChange</code>	<code>e.target.value</code>	與HTML不同，用value代表被選中的選項值(並非使用selected屬性)

## 可控的表單元素 必要條件x2

- > value屬性需要對應到某個State(狀態)值
- > 事件callback函式必需可以更動到value屬性的對應State(狀態)值

```
1 function ControlledInput() {  
2   const [data, setData] = useState('')  
3  
4   return (  
5     <>  
6       <input  
7         type="text"  
8         value={data}  
9         onChange={(e) => setData(e.target.value)}  
10      />  
11    </>  
12  )  
13 }
```

value 屬性需要對應到某個狀態

更動事件的callback必需設定value對應的狀態值