

Props(屬性)

固定的屬性值，來自父母元件或是預設值

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
...



Props(屬性)

寬度
高度
預設文字
...

Props(屬性)

父母元件對子女元件的溝通方式
(資料傳遞方式)

Props(屬性)值是一種物件資料，
可以透過函式的傳入參數得到

```
function Child(props) {  
  return <h1>{props.text}</h1>  
}
```

```
function Parent() {  
  return <Child text="今天開始學React!" />  
}
```

在父母元件使用了子女元件，並設定了屬性(Props)值

React 中只有定義了單方向的資料流動，也就是只有
父母元件能傳遞資料給子女元件(P->C)

Props(屬性) - 預設屬性

預設屬性，當父母元件沒給定值時會應用

```
function App() {  
  return (  
    <>  
      <ShowName />  
      <ShowName firstName="Amy" lastName="Yi" />  
    </>  
  )  
}
```

```
function ShowName(props) {  
  return (  
    <h1>  
      {props.firstName}, {props.lastName}  
    </h1>  
  )  
}
```

```
// 設定預設屬性(當父母元件沒給定時應用)  
ShowName.defaultProps = {  
  firstName: 'Eddy',  
  lastName: 'Chang',  
}
```

Props(屬性) - 屬性類型

PropTypes需要額外安裝套件，用於限制傳入屬性的類型(只有警告訊息)

```
import PropTypes from 'prop-types'

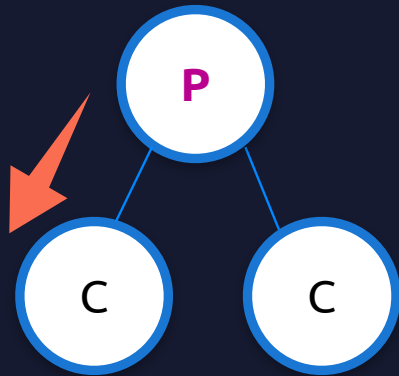
function ShowName(props) {
  return (
    <h1>
      {props.firstName}, {props.lastName}
    </h1>
  )
}

// 設定屬性的限制類型或必填等等
ShowName.propTypes = {
  firstName: PropTypes.string.isRequired,
  lastName: PropTypes.string.isRequired,
}
```

套件安裝指令：
yarn add prop-types

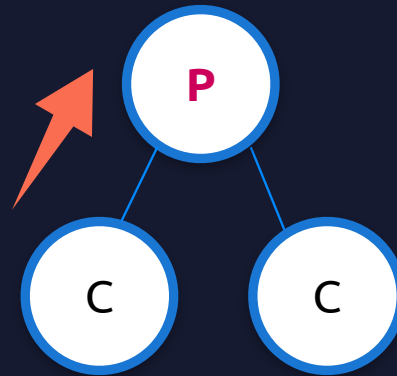
Props(屬性)

三種基本的元件溝通(資料傳遞)方式



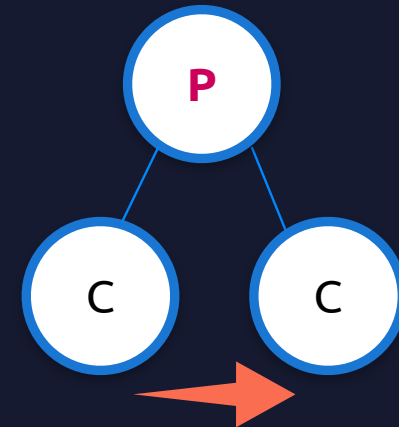
父母 -> 子女

原本就有的單向資料流



子女 -> 父母

利用callback函式



子女 -> 子女

利用父母元件與前述兩種方式

State(狀態) vs Props(屬性)

	State(狀態)	Props(屬性)
用途	在元件內部控制目前的私有資料	在元件的層級間傳遞資料
資料類型	物件	物件
在此元件中可否改變	可(讀+寫)	不可(唯讀)
可否被其它元件存取	不可	可
可否被父母元件改變	不可	可

類別型元件 vs 函式型元件

	類別型元件	函式型元件
主要語法	類別 (Class)	函式 (Function)
狀態	可以使用	可以使用 (useState勾子)
生命周期方法	有	模擬 (useEffect勾子)
優點	語法明確嚴謹/穩定	語法較為簡單
效能	相同	相同