

# 副作用 & 純函式概念

# 純函式是什麼？

- ◆ 當一個函式是純函式時，我們可以說`輸出`只取決於`輸入`
- ◆ 對於函式來說，具有副作用代表著可能會更動到外部環境，或是更動到傳入的參數值。  
函式的區分是以 純(pure)函式 與 不純(impure)函式 兩者來區分

# 純函數的條件

- ◆ 給定相同的輸入(傳入值)，一定會返回相同輸出值結果(返回值)
- ◆ 不會產生副作用
- ◆ 不依賴任何外部的狀態

# 純函式範例

純函式

```
const sum = function(value1, value2) {  
  return value1 + value2  
}
```

不純函式

```
let count = 1  
  
let increaseAge = function(value) {  
  return count += value  
}
```

# JS語言內建的純/不純函式

- ◆ `console.log`, `alert` - 會改變或依賴外部環境，為不純函式
- ◆ `Math.random` - 每次調用都會產生不同值，為不純函式
- ◆ 時間、I/O、Ajax、數據庫相關 – 全部都是不純函式

# 純函式的特殊優點

- ◆ 代碼閱讀性提高
- ◆ 較為封閉與固定，可重覆使用性高
- ◆ 輸出/輸入單純，易於測試、調試
- ◆ 可作緩存或記憶處理，在高花費的應用中可作提高運行性能的機制

## 副作用的程度差異

- ◆ 輕度/微量 – console.log/alert/Date/random
- ◆ 中度/通常 - **Ajax/Fetch API/Timer**
- ◆ 重度 – Promise/Generator或其它多個同步與異步組合控制流程
- ◆ 隱性/不經意 – 值相等運算、隱性轉類型等等