常用事件處理

透過滑鼠事件控制網頁元素

事件偵聽

.addEventListener

新增事件偵聽

```
HTMLElement.addEventListener('事件', 函式)


實際用例：
// 「event 事件參數」為事件偵聽函式自帶的 callback 參數
$btn.addEventListener('click', function(event) {
  console.log(event);
})
```

.removeEventListener

移除事件偵聽

```
HTMLElement.removeEventListener('事件', 函式)


實際用例：
// 如需使用到移除事件偵聽，須先建立好函式
const handleEvent = function(event) {
    console.log(event);
};
$btn.addEventListener('click', handleEvent);
$btn.removeEventListener('click', handleEvent);
```

滑鼠點擊事件

click

說明

左鍵點擊時觸發

dblclick

說明

短時間內雙擊左鍵觸發

mousedown

說明

任一滑鼠按鍵按下時觸發

mouseup

說明

任一滑鼠按鍵放開時觸發

滑鼠移動事件

mousemove

說明

滑鼠移動時觸發

mouseenter

說明

滑鼠進入元素邊界時觸發

mouseleave

說明

滑鼠完全離開元素時觸發

mouseover

說明

滑鼠經過不同元素時觸發，只要滑鼠經過任一子元素後，事件將被再次觸發

mouseout

說明

滑鼠離開元素時觸發，只要滑鼠經過任一子元素後，事件將被再次觸發

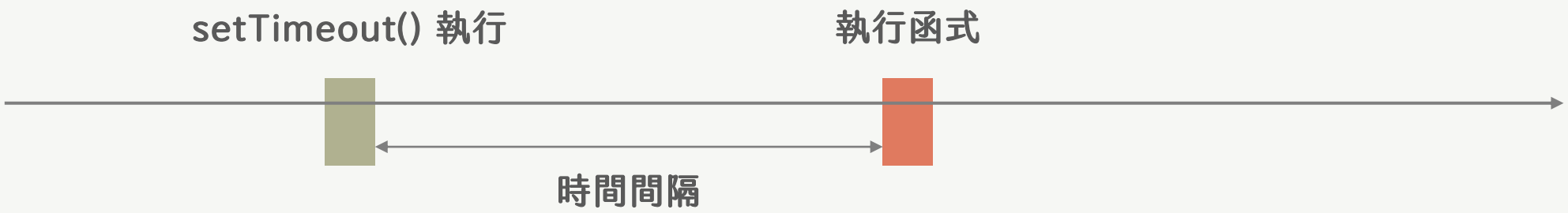
專案實作

製作計數器

時間函式基礎

setTimeout / setInterval / clearInterval

setTimeout(函式, 時間)



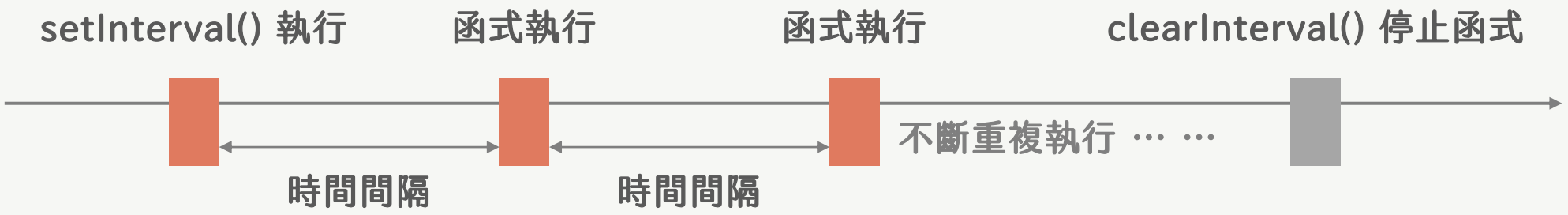
說明

於指定時間（單位為毫秒）後，執行「一次」指定的程式碼

實際用例

```
setTimeout(function() {
  console.log('hello world')
}, 3000)
```

setInterval(函式, 時間)



說明

於指定時間隔間（單位為毫秒）重複執行至 clearInterval() 執行為止

實際用例

```
let percent = 0
const timer = setInterval(function() {
  percent++
  if (percent > 100) {
    clearInterval(timer)
  }
}, 100)
```