



4 函式 Function

函式是一連串的程式碼集中在一個區域中。這一連串的程式碼“只有”在被呼叫時才會執行。函式可以讓我們在分段區中妥善的組織程式碼，提高程式碼的重覆利用性。

要使用函式只需要兩部份 (1) 宣告函式，以及 (2) 使用/執行函式

函式名稱

給定函式的識別名稱。

技巧: 描述你的函式名稱就像它要作什麼事情一樣。

回傳 Return (可選的)

函式可以自訂在當它被呼叫後回傳一個值。當函式回傳時，接下的程式碼就不會再繼續執行。

呼叫函式

呼叫(Invoking, calling)或執行(running)函式都是同一個意思。當我們撰寫了一個例如名為 `someName` 的函式，後面加上圓括號符號 `()` 後，像是 `someName()`，在函式中的程式碼就會被執行。

// 函式定義/陳述式

```
function someName(param1, param2) {
```

// 一連串所需的程式碼...

```
var a = param1 + "love" + param2;
```

```
return a;
```

```
}
```

// 呼叫(run/call/invoke)函式

```
someName("Me", "You")
```

傳入參數(Parameters / Arguments) (可自訂的)
函式可以自訂是否要接受傳入參數 (稱為parameters或稱為arguments)。函式可以之後在程式碼中使用這些資訊。

程式碼區塊
任何包含在花括號 `{ ... }` 中的程式碼稱為“程式碼區塊”。這個概念不只是函式有，像“if 陳述式”，“for loops”與其它的陳述式都有使用程式碼區塊的方式。

傳遞參數(parameter)到函式 (可自訂的)
在呼叫函式時，可以傳遞參數到函式中的程式碼。