

## 第 2 章

# 沒有註解？真令人無法接受！

藉著使用有意義的變數名稱，我們可以讓我們的程式碼更具可讀性並更容易理解 [1]。

[1]

```
float pictureWidth, pictureHeight, pictureSurfaceArea;
pictureWidth = 8.0;
pictureHeight = 4.5;
pictureSurfaceArea = pictureWidth * pictureHeight;
```

不同於我們目前所見的範例中那短短的幾行，您真正撰寫的程式碼會越來越長。撰寫程式時主要的重點並不只是擔心它是否確實的照著您所要的動作，還要確保它擁有良好的文件。之後，如果您有段時間沒有看這些程式碼卻要修改它，您真的需要註解來協助您了解您的某段程式碼到底是用來作什麼的，以及這個部份當初為什麼在那邊。我們強烈的建議您花些時間為您的程式碼加上註解，並保證未來您將會從許多方面贏回花費在這上面的時間。同樣的，如果您分享您的程式碼給別人，只要您有提供註解，對方就可以很快的修改以符合他自己的需求。要下一個註解，請直接在該註解的前面輸入兩個斜線。

```
// 這是註解
```

在 Xcode 中，註解是綠色的。如果註解很長，並且跨越了多行，請把它放到 /\* 與 \*/ 之間

```
/* 這是一個註解
而且跨越了兩行 */
```

雖然 Xcode 對除蟲工作大有助益，但我們還是簡短的說明如何為程式除蟲。一個古老而過時的除蟲方法稱為註解法（outcomment），藉著把部份的程式碼放到 /\* \*/ 之間，您可以暫時的停用（註解掉）這部份的程式碼來看看其餘的部份是不是照計畫的運作。這個方式可以讓您用來獵殺臭蟲，如果被註解起來的部份要，例如說，指定一個值給某個變數，您可以加一行暫時的程式碼來把適合的值指定給此變數以測試其他沒有註解起來的部份是否正常運作。

註解的重要性無可言喻。為一長串的述句加上解釋來說明它們到底作了什麼事是很有幫助的，這是因為您不需要推測這段程式碼在做什麼，而且您可以立即查看您正遇到的臭蟲是不是在這個部份的程式碼中。您也應該利用註解來表達一些困難，或者無法由程式碼中推測出來的事項。舉例來說，如果您正設計一個數學函數並使用一個在課本中詳細描述的特殊模型，您應該在實際的程式碼旁邊用註解註明參考書目。有時候在實際撰寫程式碼前先寫下註解是有幫助的。它會協助您理清思緒並讓程式設計變得較為容易。

本書的範例並沒有像我們平常撰寫時一樣多的註解，因為它們已經環繞在大量的說明之下。