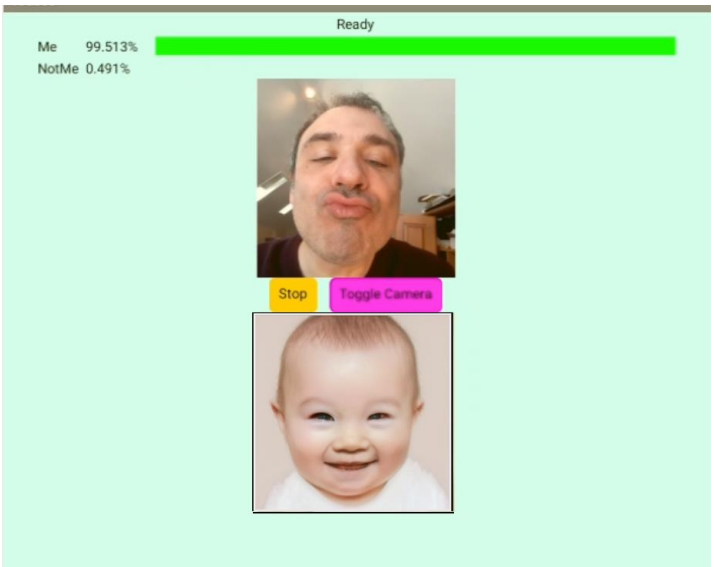
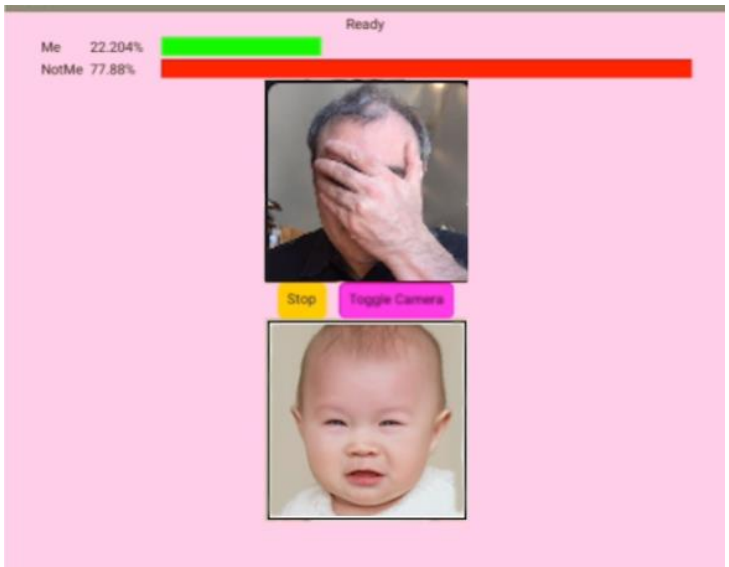
躲貓貓：第二部分





Peekaboo 遊戲

初生不久的嬰兒還不太了解如何用視覺去理解周遭的世界。在美國，父母經常與嬰兒玩一種叫 “Peekaboo”（中譯「躲貓貓」）的親子遊戲：父母遮住臉（用手或枕頭等），令嬰兒驚訝父母「消失」了：然後父母又突然向嬰兒展示臉容說：“Peekaboo, I see you!” 嬰兒因為看到父母又突然「橫空出現」而被逗樂。

試試使用老師為你提供的 PICaboo 程式。用手顯示和隱藏你的臉，看看程式中的嬰兒如何反應。

躲貓貓：第二部分

從這裏開始

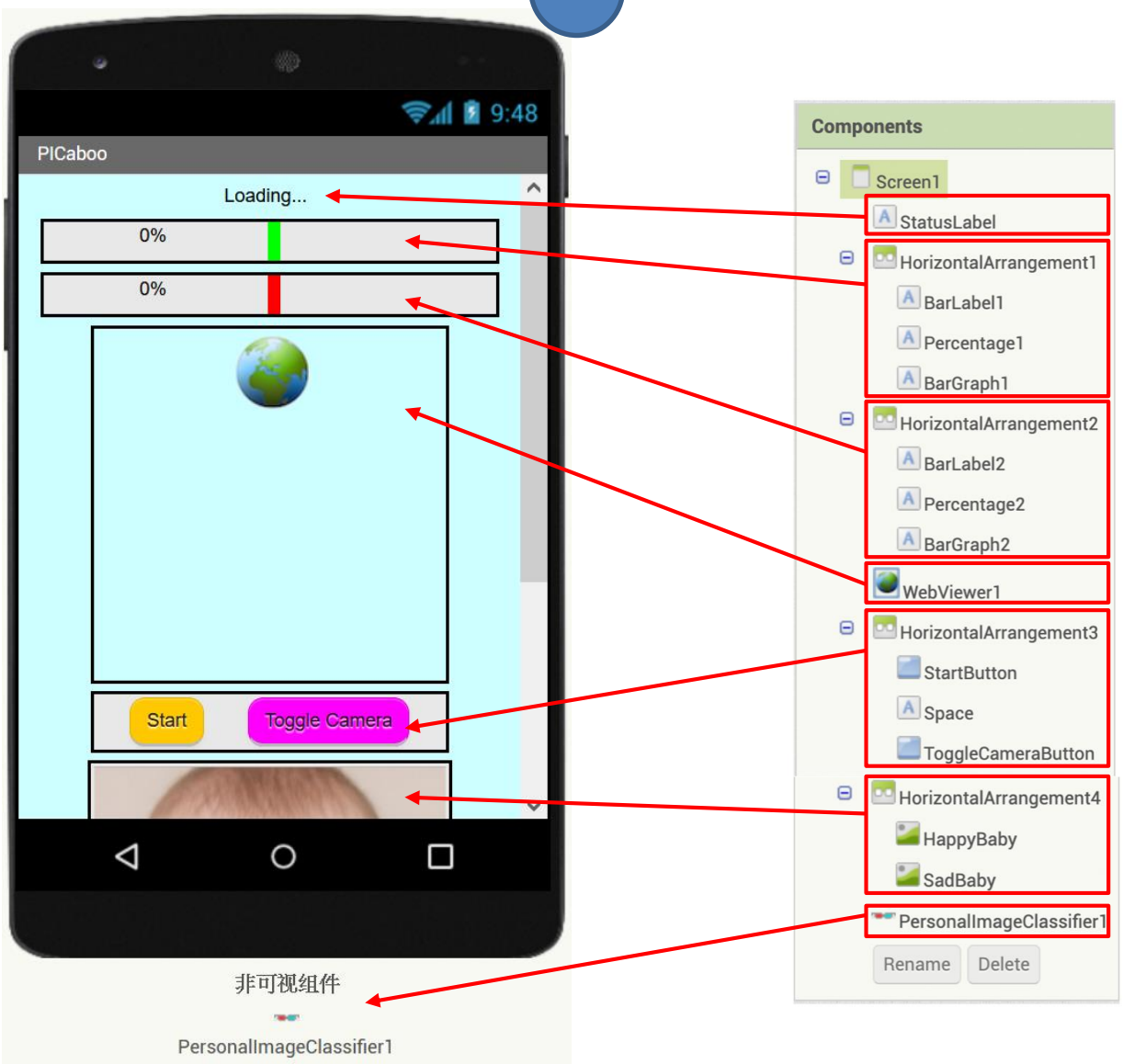
你在本教節中會使用 Personal Image Classification (PIC) 的模型，製作一個 Peekaboo 遊戲的應用程式，該程式中的嬰兒會在你露臉時微笑，在掩臉時哭泣。

❑ 前往 http://ai2.appinventor.mit.edu，開啟由老師提供的範例檔。

1

❑ 以下是「畫面編排」(Designer) 內已包含的元件。試查看每個元件的用途。

2

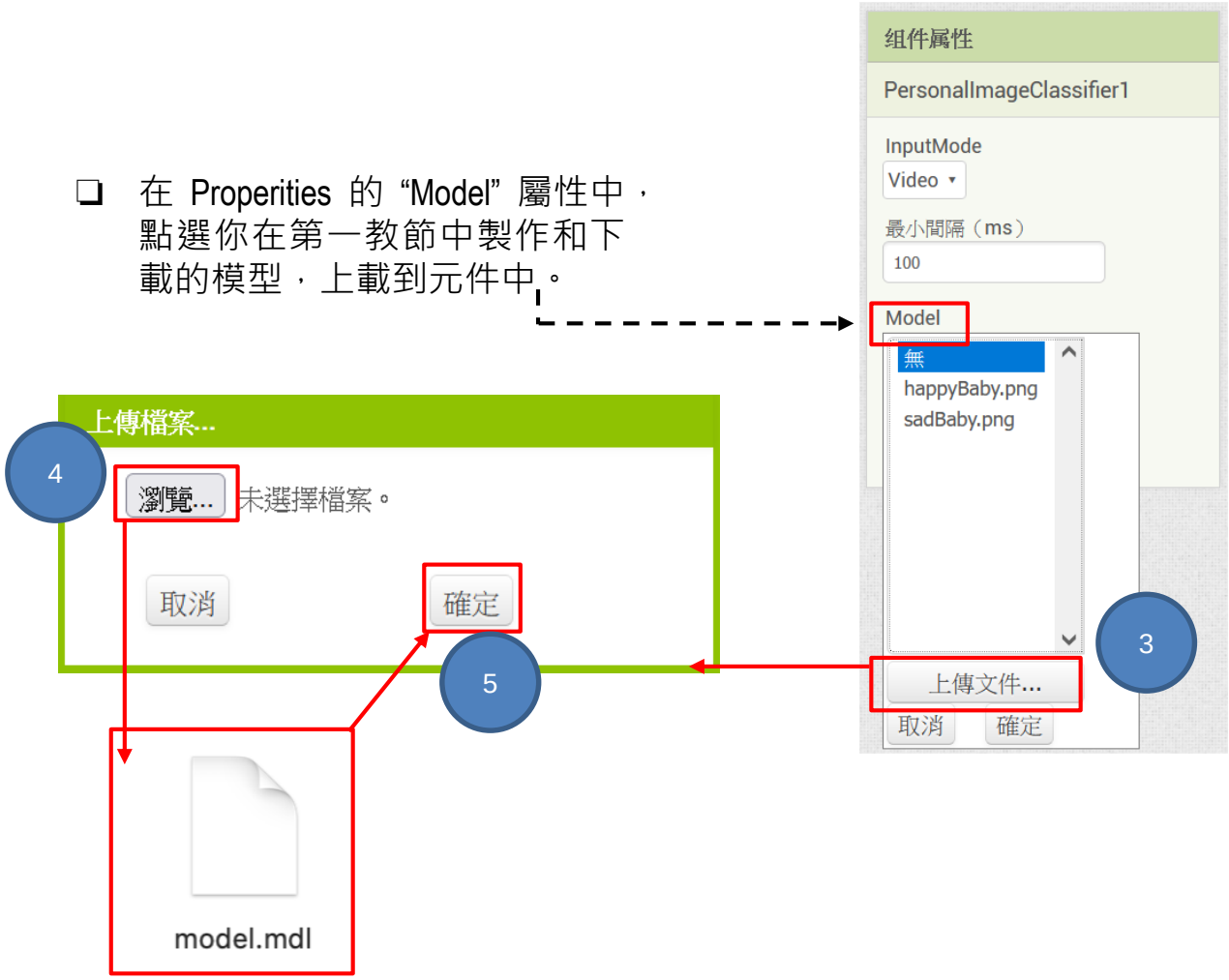


躲貓貓：第二部分

加入模型到 PERSONALIMAGECLASSIFIER

- 看看「非可視元件 **PersonalImageClassifier1**」，它可根據你訓練好的模型，對你的臉容進行分類。

- 在 Properities 的 “Model” 屬性中，點選你在第一教節中製作和下載的模型，上載到元件中。



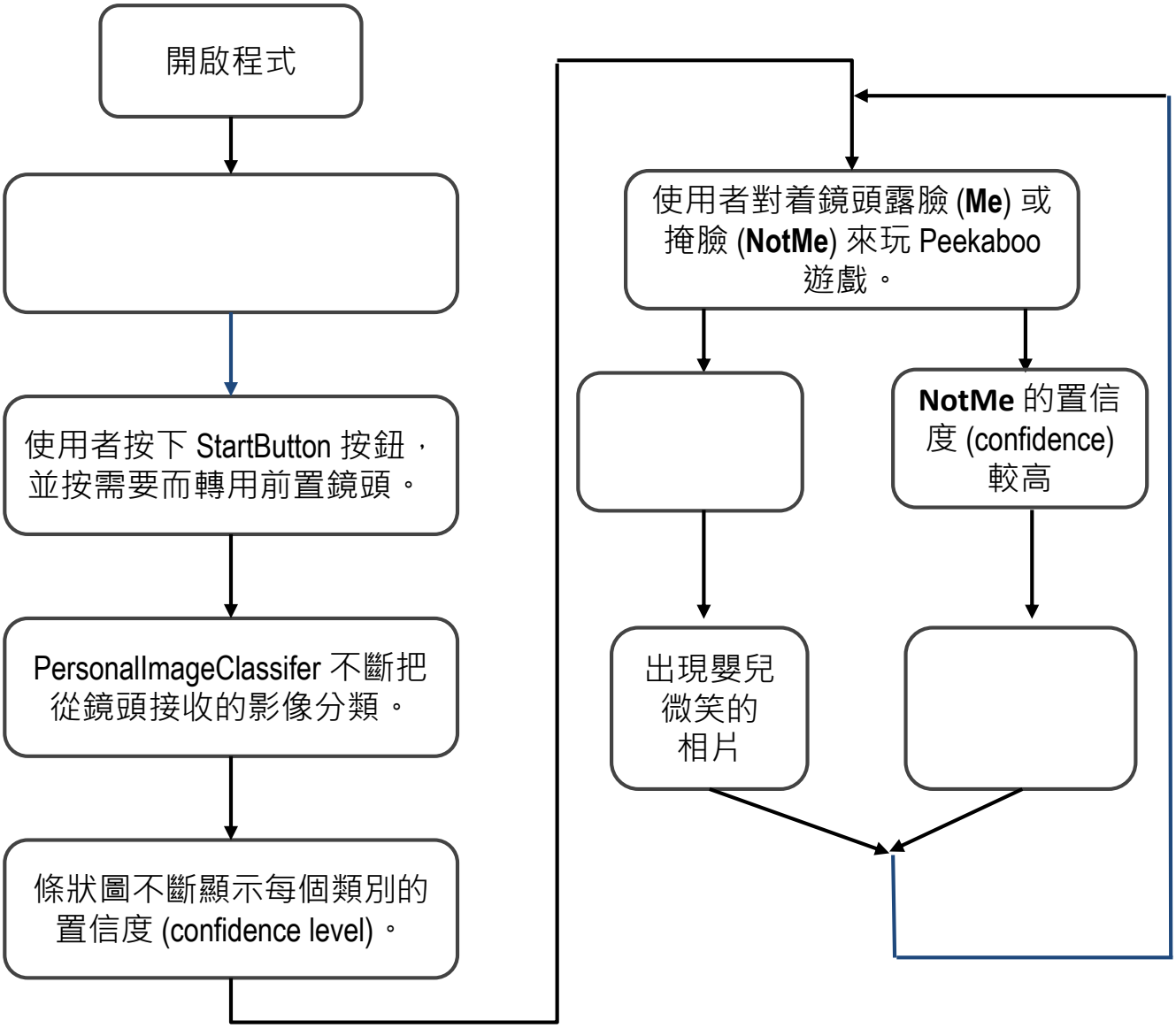
躲貓貓：第二部分

<u>事件的順序</u>

試循着以下 PICaboo 程式中運行的事件順序，把相應的字母填到空格中。

6

A. 程式顯示 “Ready”	B. **Me**的置信度 (confidence) 較高	C. 出現嬰兒哭泣的相片



躲貓貓：第二部分

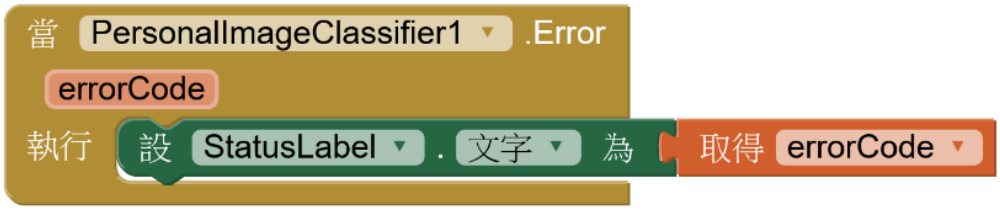
檢閱預設指令方塊

❑ 現在切換到「程式設計」(Blocks) 來開始編程。

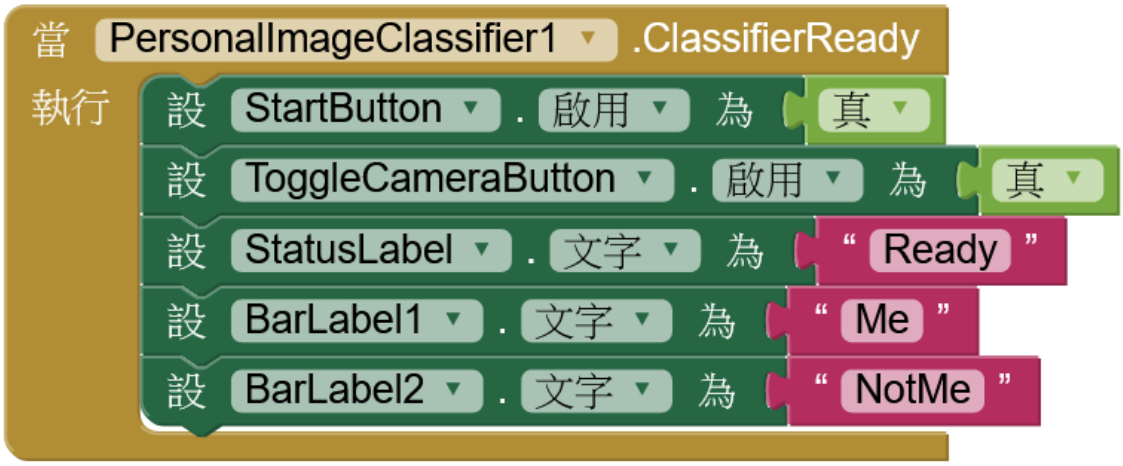
7

❑ 看看部分指令方塊已經預先設定在範例中。它們組成了 **PersonalImageClassifier**，並在應用程式啟動時首先運作。

萬一 **PersonalImageClassifier** 出現錯誤，相關的事件類指令方塊會在 **StatusLabel** 顯示錯誤。



當準備好為影像分類時，會呼叫 **when PersonalImageClassifer.ClassifierReady** 指令方塊，並啟動 **StartButton** 和 **ToggleCameraButton** ，而 **StatusLabel** 會顯示 “Ready” ，**BarLabel1** 和 **BarLabel2** 也會分別標上模型的標籤名稱（**Me** 和 **NotMe**），即是在模型中經過訓練的類別。

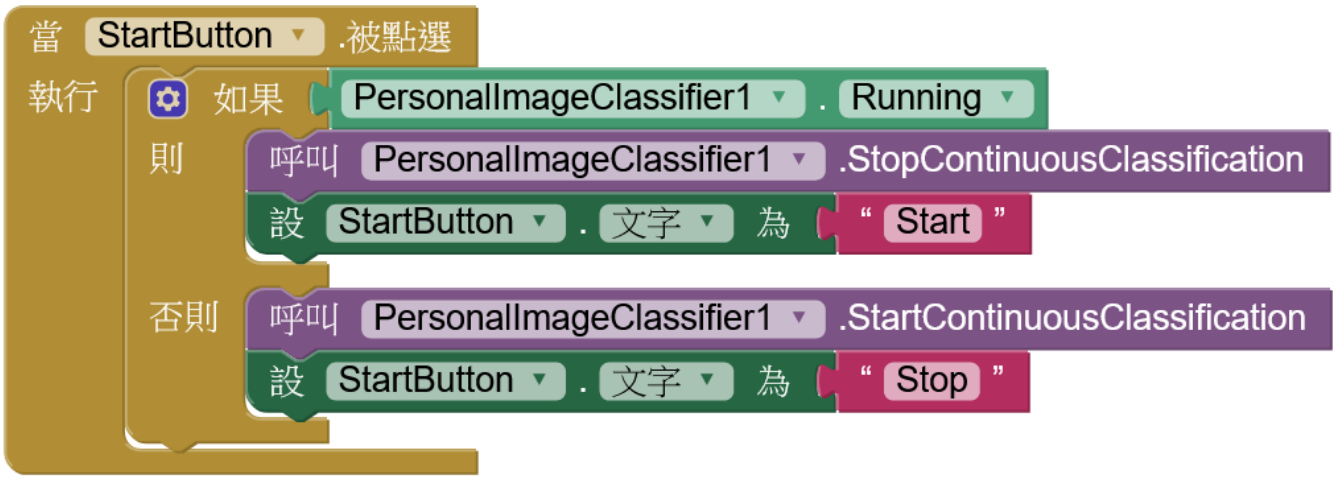


躲貓貓：第二部分

<u>檢閱預設指令方塊</u>

- 當按下 **ToggleCameraButton** 按鈕，便會從前置鏡頭切換到後置鏡頭（或相反）。

- 按下 **StartButton** 按鈕會切換為「開始」或「停止」：如果按下時，**PersonalImageClassifier1** 元件正在運作，便會中止其影像分類工作；否則的話，便會啟動其工作。按鈕上的文字亦會隨之而切換成 “start” 或 “stop”。



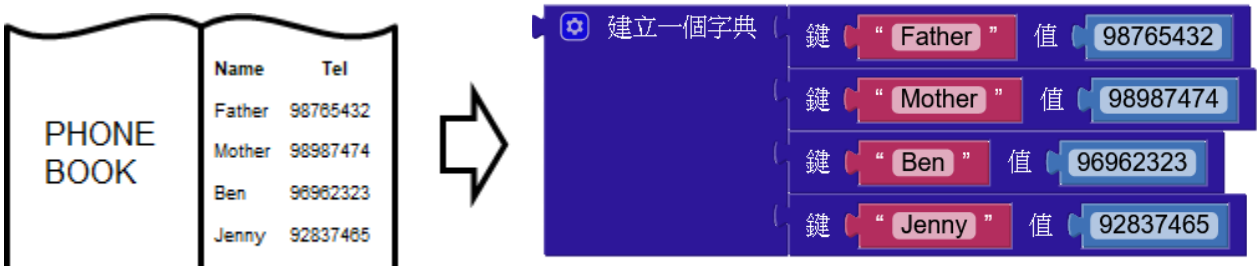
躲貓貓：第二部分

影像分類的結果

在正式為 PICaboo 編程之前，我們來看看讓我們看看如何接收和使用來自 **PersonelImageClassifier1** 元件的分類結果。

- 當 **PersonalImageClassifier1**元件為鏡頭拍攝到的影像完成分類時，會呼叫 **when PersonalImageClassifier1.GotClassification** 事件指令方塊。這也是你在本教節中的主要任務：完成處理這個事件的指令方塊。

- **result** （結果）以 dictionary（字典）的數據結構的形式列出分類，即是一系列「鍵—值對」(key-value pairs)。詳見下例：



有關這些「鍵—值對」的數據結構方式，可以用 **Dictionary** 指令方塊呈現出來。

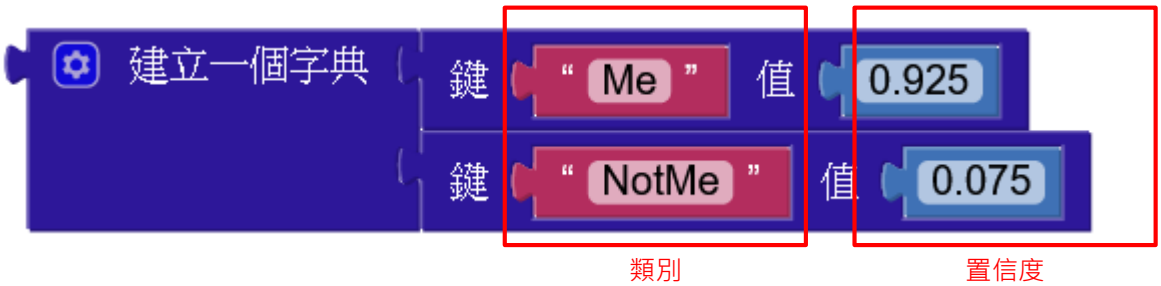
- 在以上「鍵—值對」的例子中，Name 是「鍵」(key)，Tel 是「值」(value)，整個電話簿 (PHONEBOOK) 則是「字典」(dictionary)。

- 在 PICaboo 程式中，**when PersonalImageClassifier1.GotClassification** 指令方塊在 **result** 中傳回的「鍵—值對」清單數據又是甚麼？

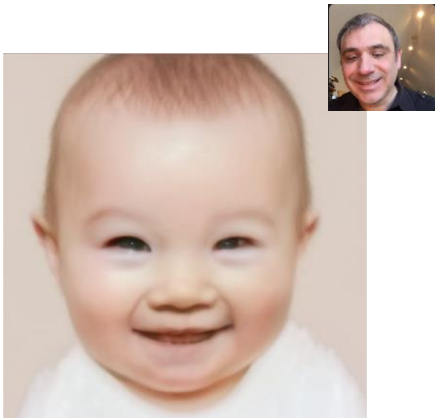
躲貓貓：第二部分

DICTIONARY 數據結構

- 由 PersonelImageClassifier1 傳回的 **result**（結果）列出了兩個類別（“**Me**” 和 “**NotMe**”）的名稱 (key) 及其置信度 (value)。相關的數據結構可用以下的指令方塊表達出來。留意以下 **result** 的「鍵—值對」是由最高到最低順序列出其置信度的。



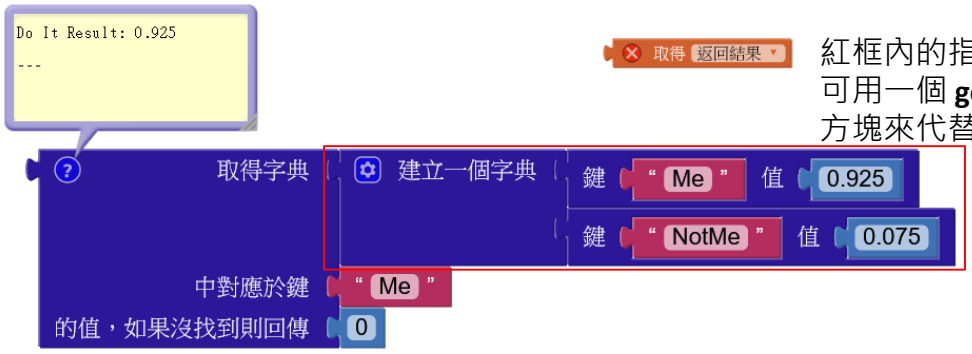
- 例如，有 **result** 傳回的數據顯示：類別 **Me** 的 置信度較高（0.925，即 92.5%），而類別 **NotMe** 則較低（0.075，即 7.5%）。這些數字都是由你在第一教節中訓練出來的模型所運算出來的。根據上例，PICaboo 程式會因而出現一張嬰兒微笑的相片，因為嬰兒（機器）確信「看到」你露出臉來！



躲貓貓：第二部分

影像分類結果

- 為了要從 **result** 中找到指定「鍵」(key) 的相關「值」(value)，就要用上 **get value for key** 指令方塊。如果你要在下例中，從指定「字典」搜尋「鍵」“Me”，就會獲得 “0.925” 這個相關的「值」（即 **Me** 的置信度）；如果找不到吻合的「鍵」，就會獲得 “0” 這個值。



紅框內的指令方塊組合，可用一個 **get result** 指令方塊來代替。

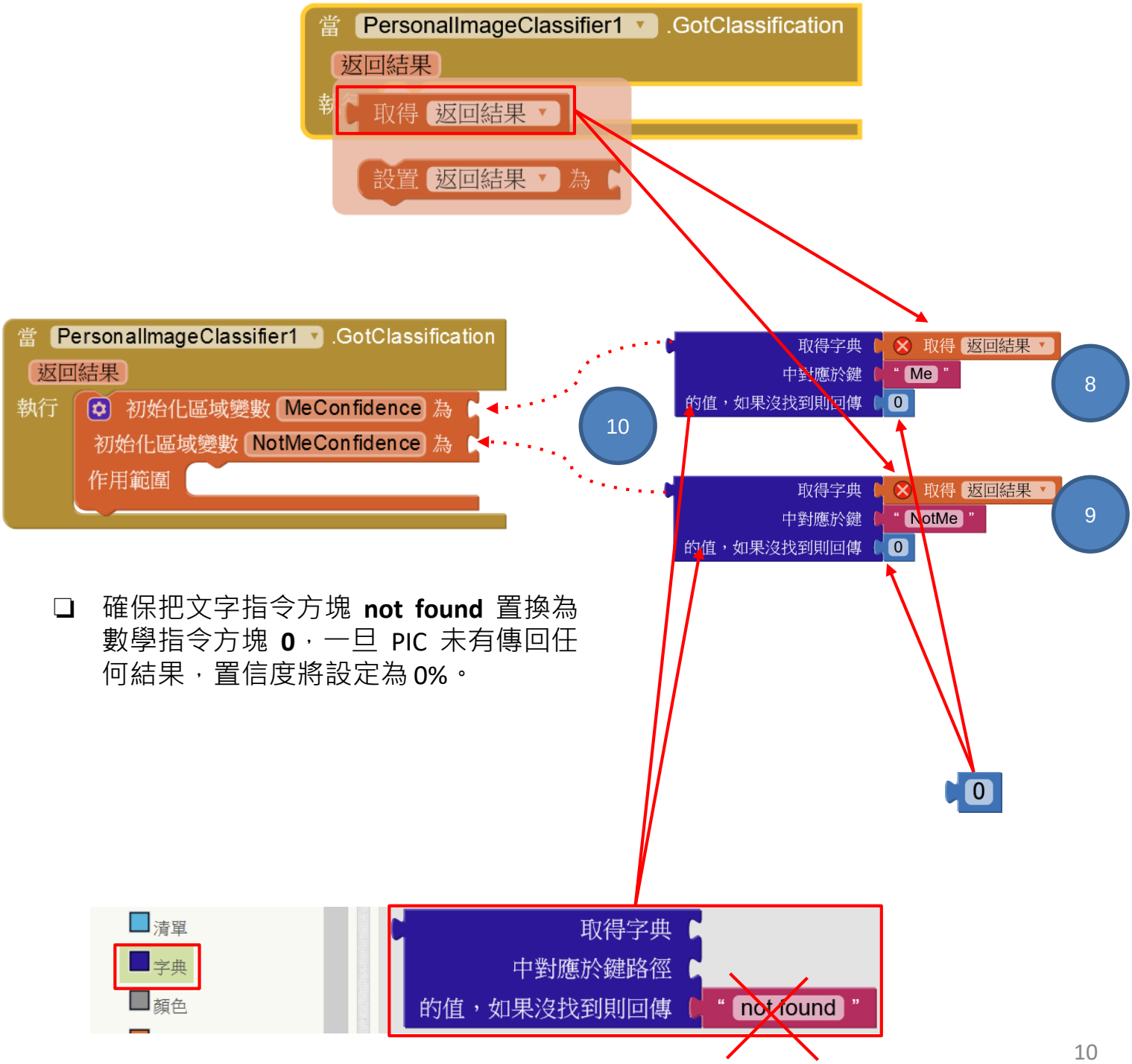
- **when PersonalImageClassifier1.GotClassification** 事件型指令方塊的組合由初始化兩個區域變數 (local variables) 開始。區域變數 (local variables) 與全域變數 (global variables) 不同，前者只能對嵌入其中的指令方塊產生效果。在這裏的區域變數定義如下：
  - **MeConfidence**:：這是 **Me** 類別的置信度（露臉）
  - **NotMeConfidence**：這是 **NotMe** 類別的置信度（掩臉）

你將會看到怎樣使用這些變數。

躲貓貓：第二部分

初始化區域變數

- 初始化下圖的兩個區域變數。如果將鼠標移到參數 **result** 上面，將可以獲得所需的指令方塊 **(get result)**。

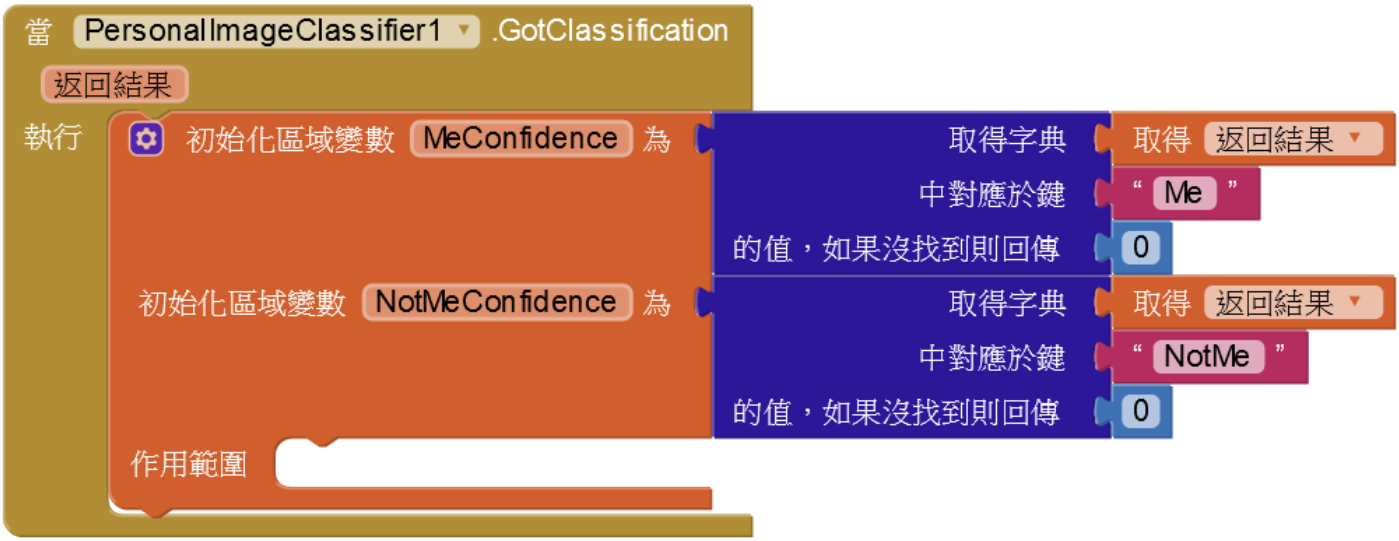


- 確保把文字指令方塊 **not found** 置換為數學指令方塊 **0**，一旦 PIC 未有傳回任何結果，置信度將設定為 0%。

躲貓貓：第二部分

<u>初始化區域變數</u>

❑ 完成後，區域變數將如下圖所示般初始化。

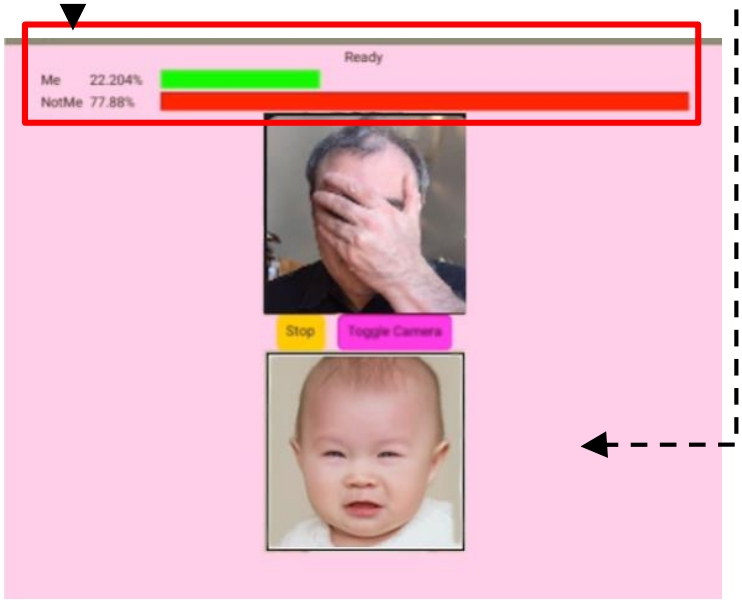


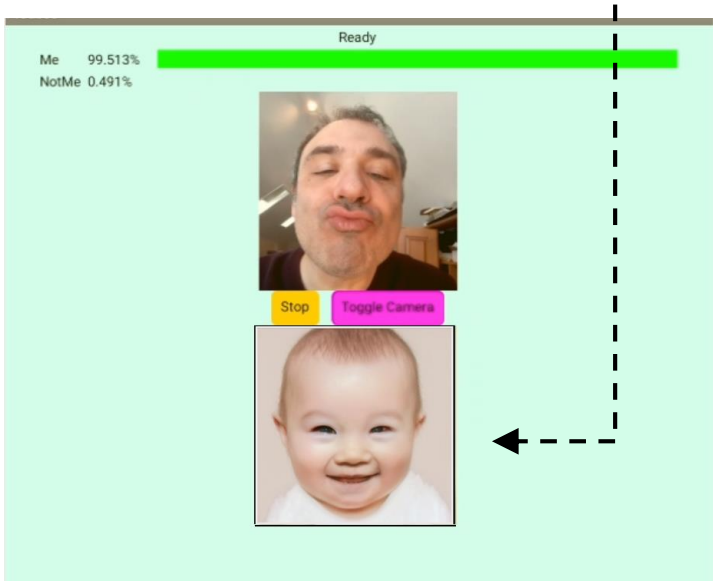
躲貓貓：第二部分

<u>事件序列</u>

❑ 當程式獲得分類數據.....

- 設定 **Percentage1** 的文字和 **BarGraph1** 寬度百分比 (%)。
- 設定 **Percentage2** 的文字和 **BarGraph2** 寬度百分比 (%)。
- 如果分類數據顯示 **Me** 的置信度較高（即是機器更有信心認為它看到你的臉） ，則展示嬰兒微笑的相片，隱藏嬰兒哭泣的相片，並把 **Screen1** 的背景顏色轉為淺綠。
- 否則，展示嬰兒哭泣的相片，隱藏嬰兒微笑的相片，並把 **Screen1** 的背景顏色轉為淺粉紅。

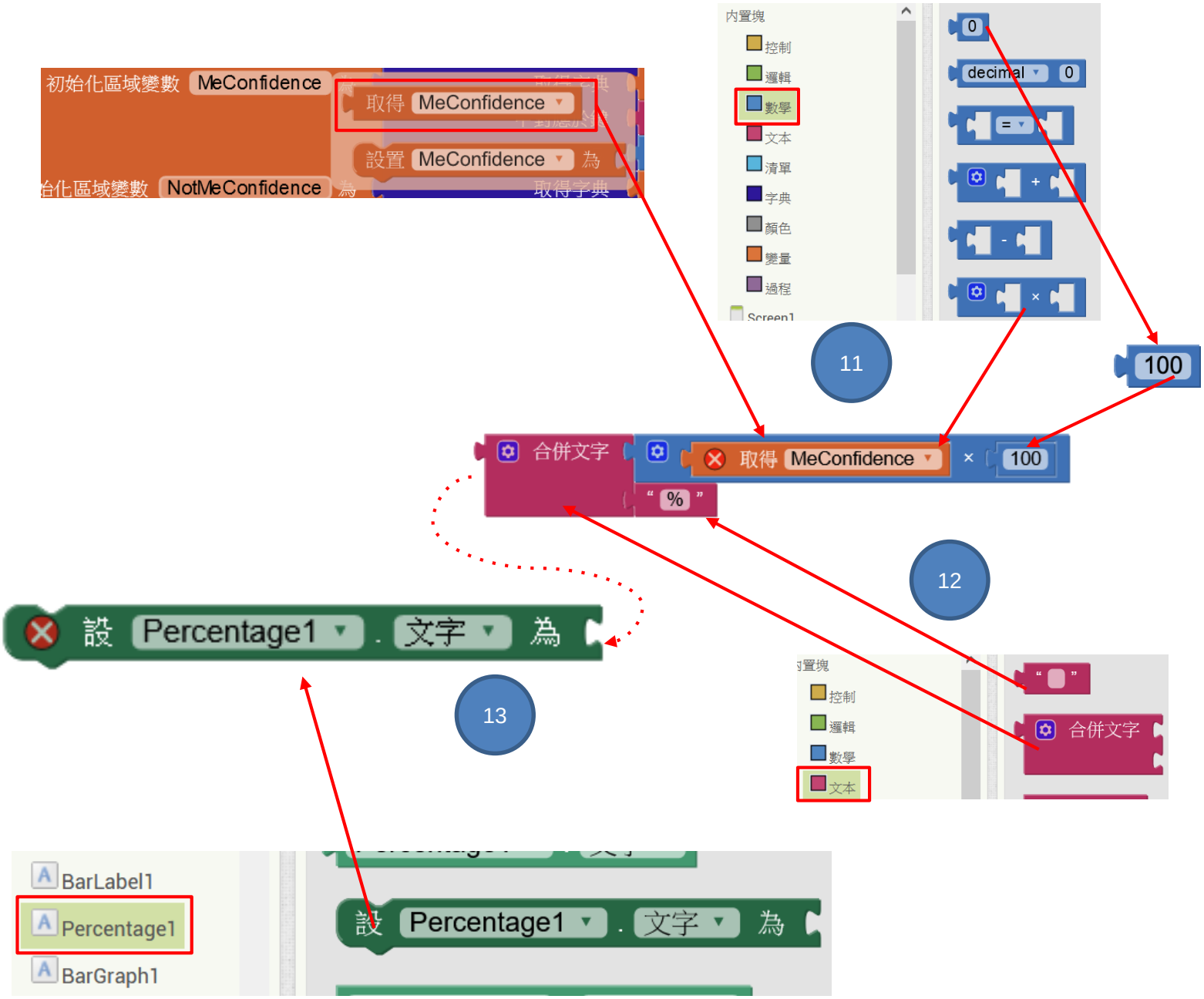




躲貓貓：第二部分

以百分率展示置信度

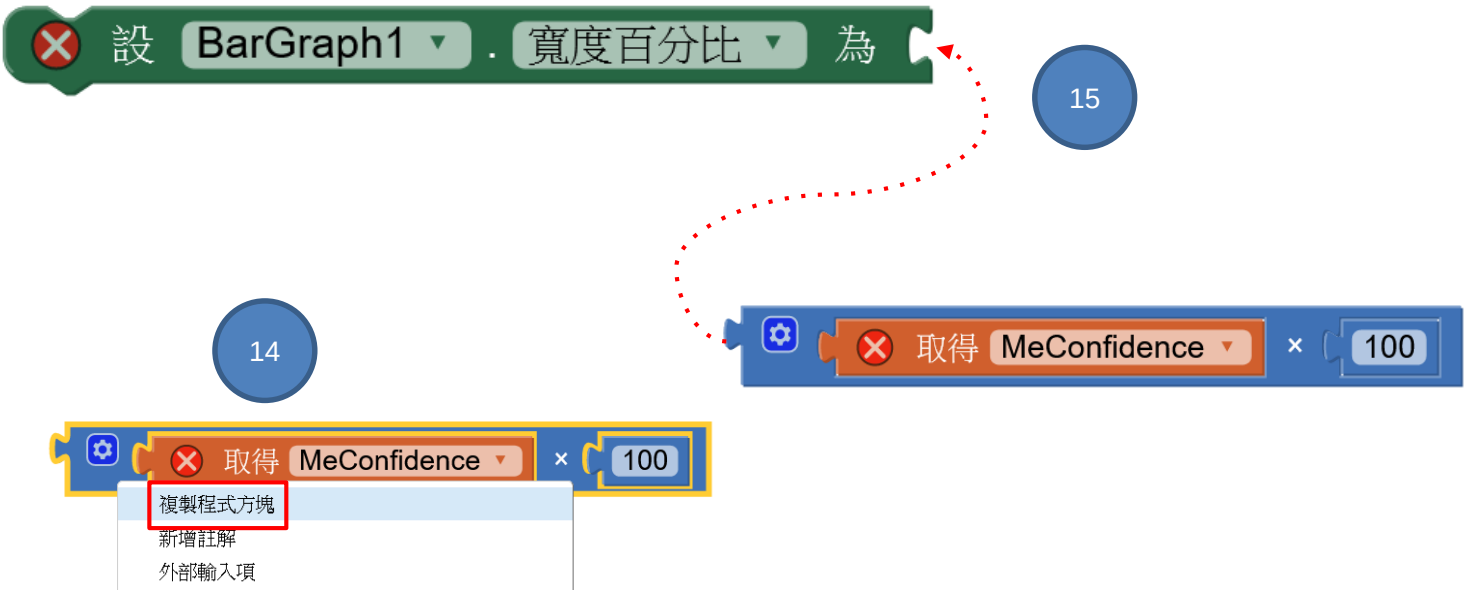
- 首先設定 **Percentage1** 的文字為 **Me** 類別的置信度。 由於該值是介乎 0 與 1 之間的小數，你要把它轉為百分率（乘以 100 後加上 “%”）。例如置信度是 0.75 的話，百分率就是 75%。



躲貓貓：第二部分

以條狀圖展示置信度

- **BarGraph1**是一個彩色標籤，其寬度的變化就是以圖形方式表示 Me 類別的置信度。試設置相關的WidthPercent，你可以複製先前編寫的部分指令方塊。

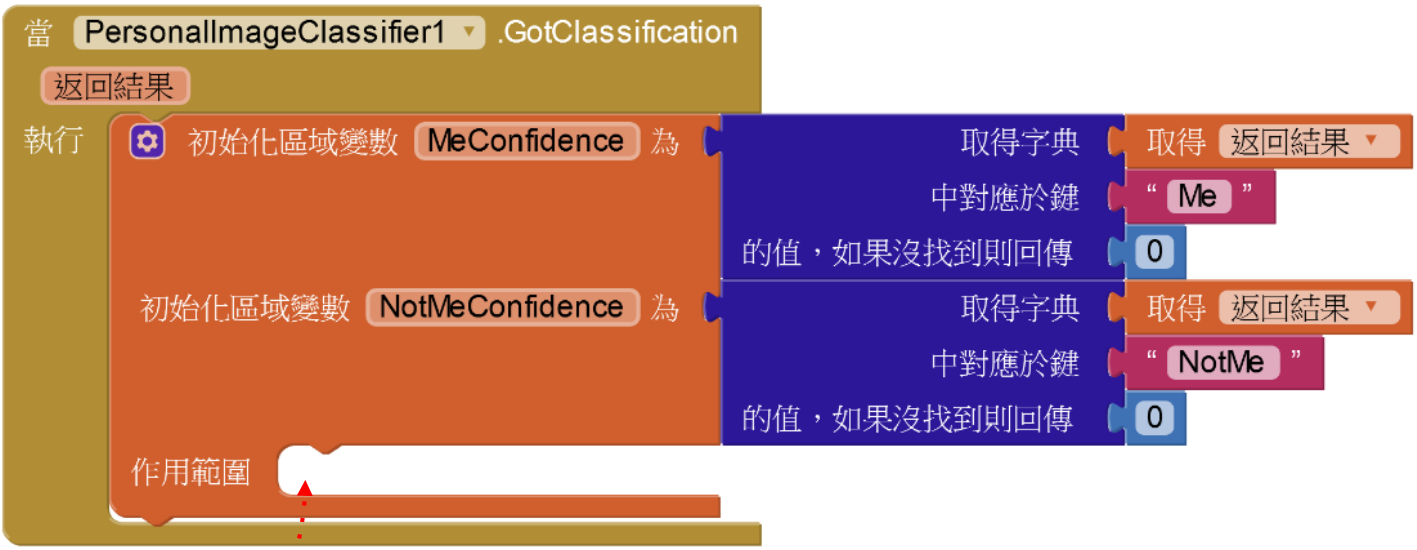


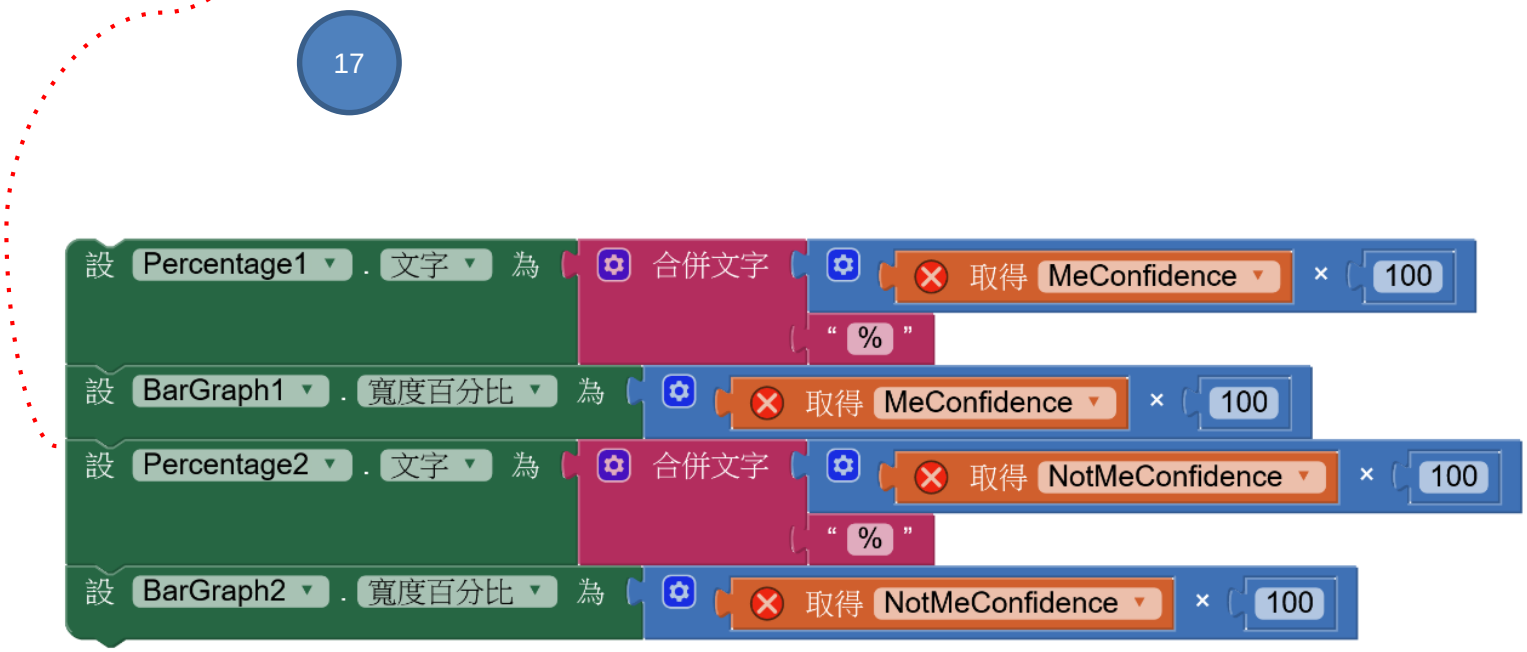
- 為 **Percentage2** 和 **BarGraph2** 作下列類似的設計。緊記以 **NotMeConfidence** 作為另一個區域變數。

躲貓貓：第二部分

以條狀圖展示置信度

❑ 把指令方塊放到以下位置。

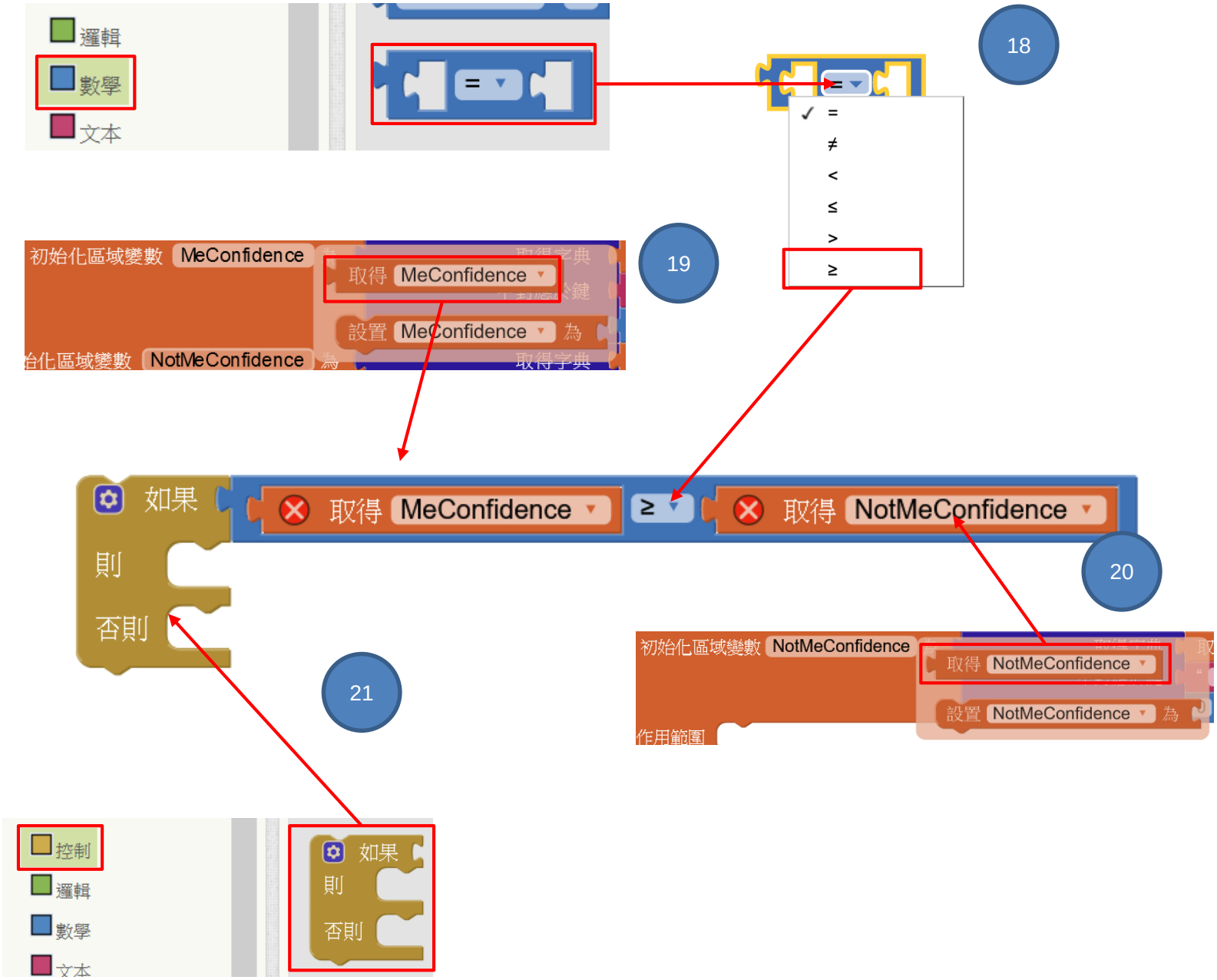




躲貓貓：第二部分

條件陳述句

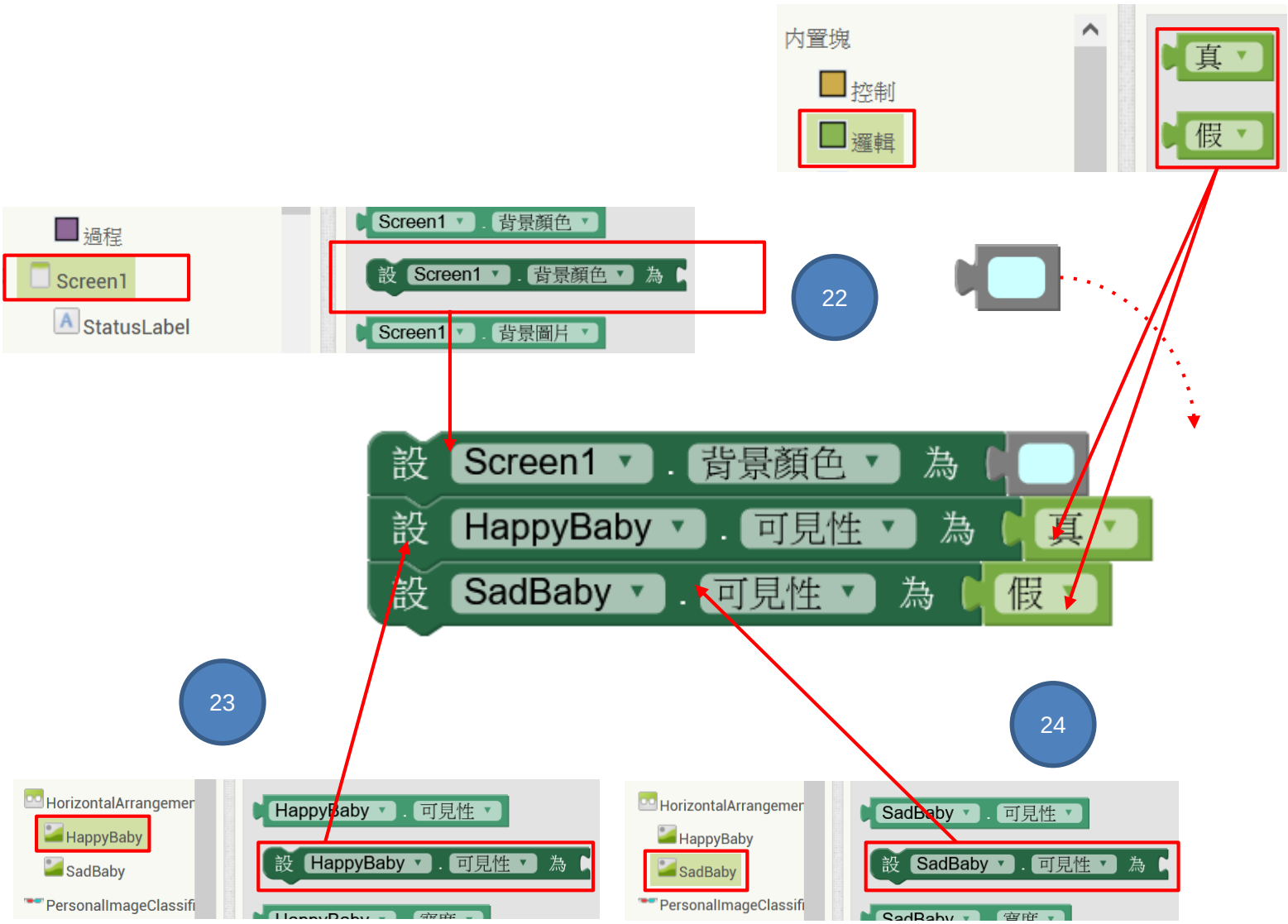
- 現在構建 **if-then-else**（如果—則—否則）條件指令方塊，來測試置信度較高的類別是否為 **Me** 類，使機器根據較高置信度，而識別出你展示了臉部。



躲貓貓：第二部分

<u>顯示嬰兒的笑臉</u>

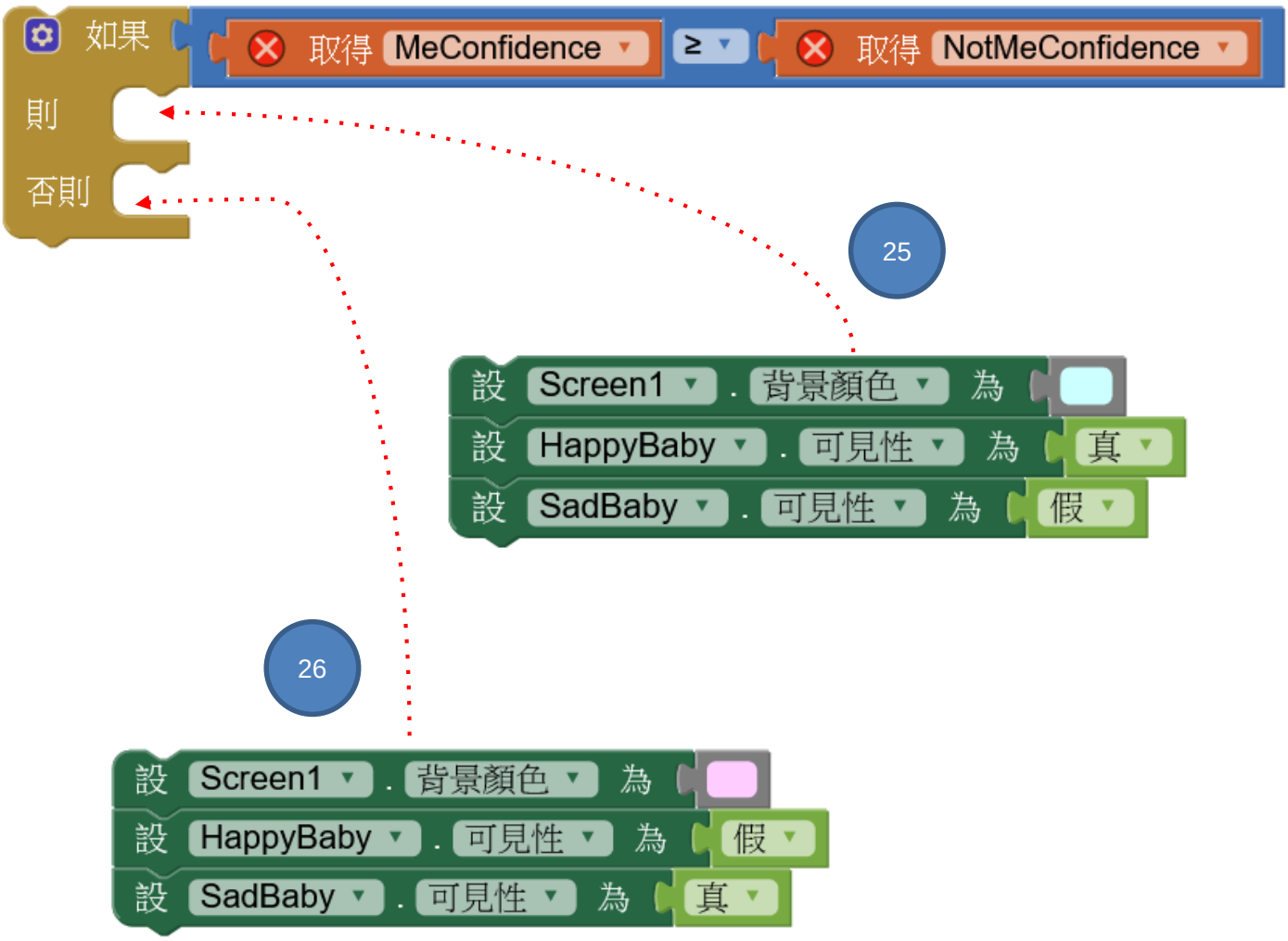
- 現在構建 **if-then-else** 條件指令方塊的 **then** 部分，你可使用預先在範例檔提供的淺綠色指令方塊。



躲貓貓：第二部分

DISPLAYING A SAD BABY

- 現在以類似方法，構建 **if-then-else** 條件指令方塊的 **else** 部分，並將這些指令方塊放到如下圖所示的位置。你可使用預先在範例檔提供的淺粉紅色指令方塊。

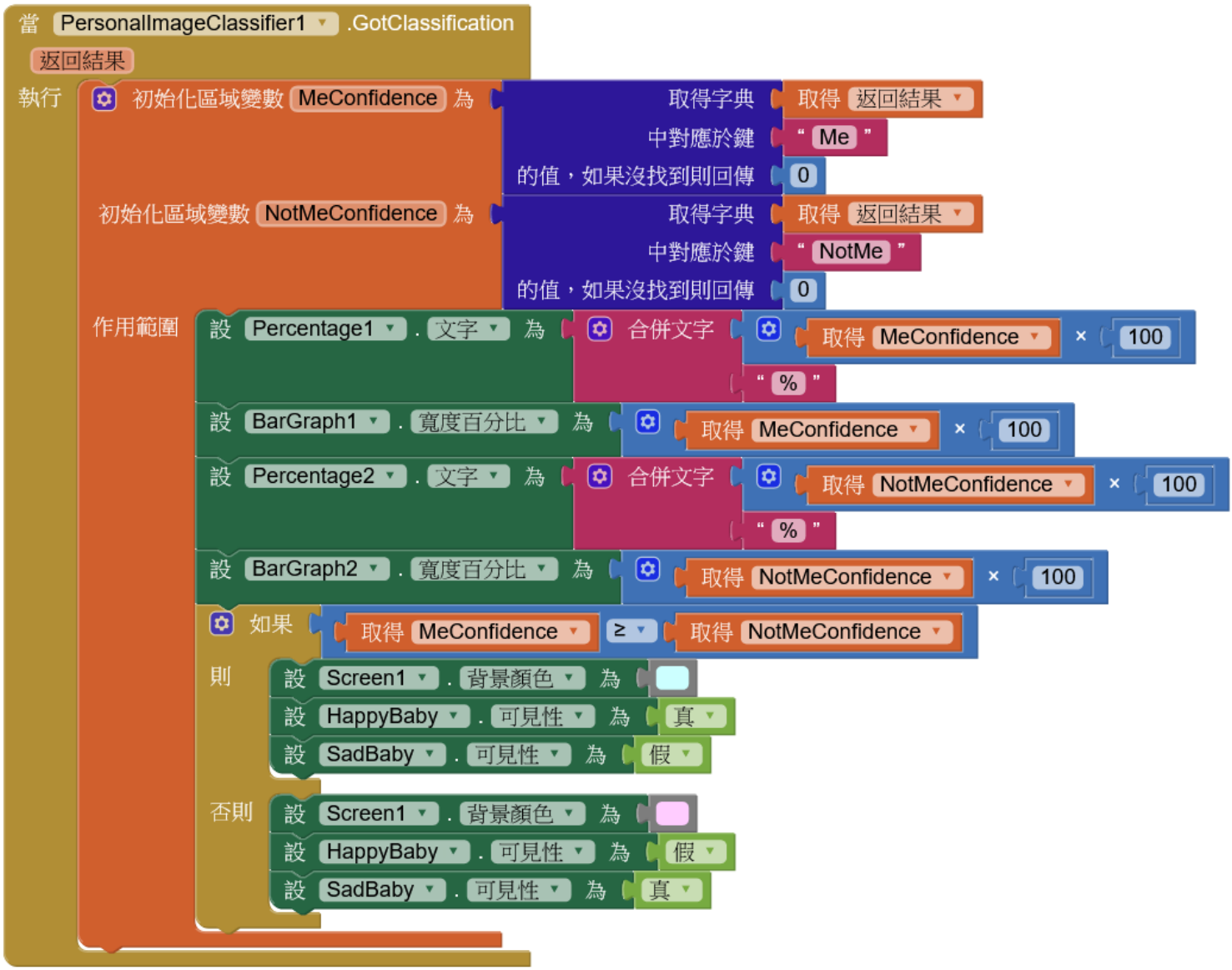


躲貓貓：第二部分

影像分類結果

❑ 完成編程後，有關 **GotClassification** 的事件型指令方塊組合應如下圖般。

27



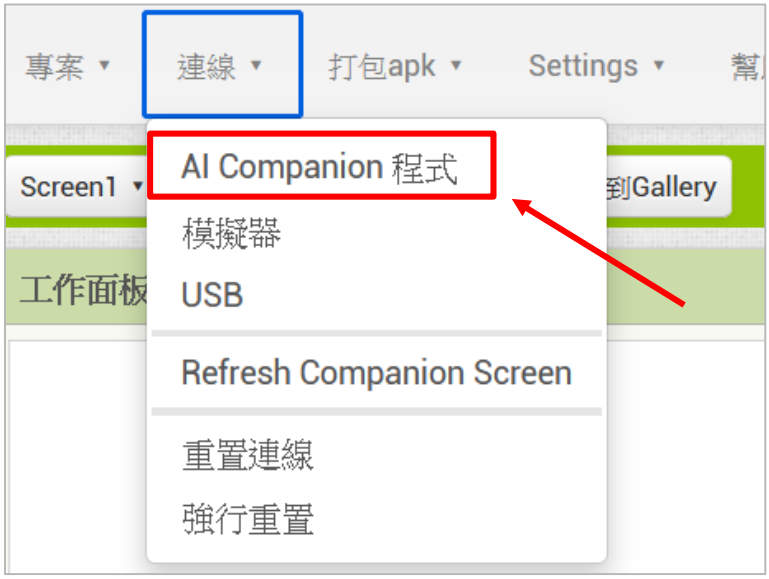
躲貓貓：第二部分

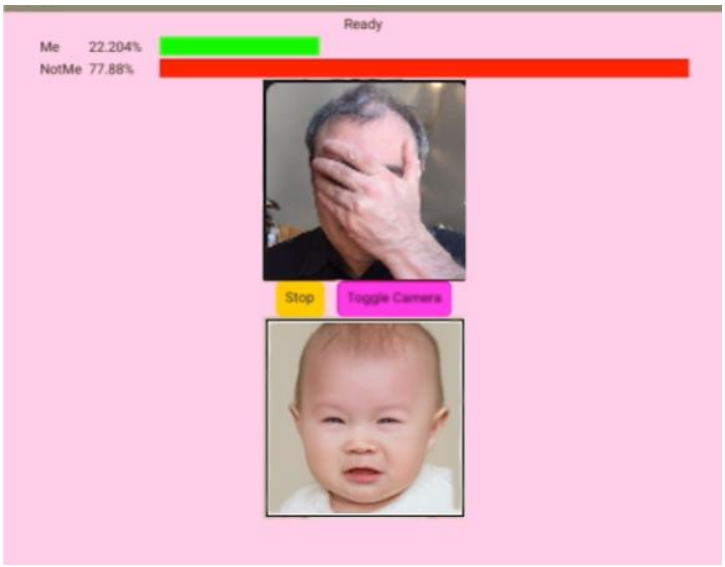
測試及除錯

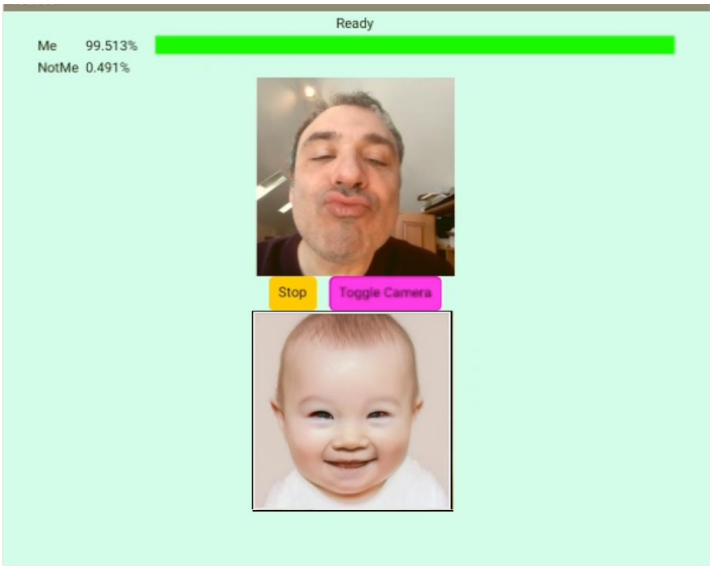
- ❑ 現在是時候測試 Peekaboo 遊戲了！試連接到 AI Companion，看看你是否可以和「嬰兒」玩 Peekaboo。緊記掩臉時要稍微張開指隙，以便看到程式發生了甚麼事。當「嬰兒」看到你的臉時會笑嗎？當你掩着臉時，「嬰兒」會哭嗎？當你在臉前上下揮動着手，反覆顯示和隱藏自己的臉時，會發生甚麼事？置信度又會怎樣？

- ❑ 恭喜你，你成功了！

28





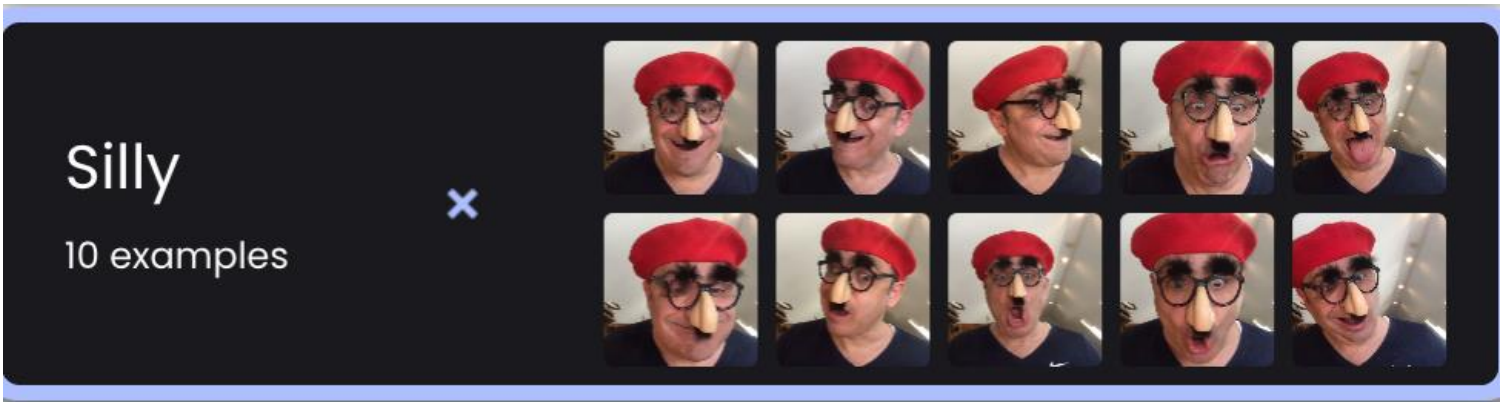


躲貓貓：第二部分

<u>延展你的專案</u>

以下主意有助延展你的專案，試試自己參考一下：

- ❑ 重新訓練你的模型，並加入除了露臉 (Me) 和掩臉 (NotMe) 之外的第三種臉容表情類別，例如傻臉 (Silly)：
  - 先備份好原本的 PICaboo 程式，再另存一個副本以作處理。
  - 上載經過了重新訓練的新模型，並修改應用程式為三個條狀圖 (bar graph)，來分別顯示三個標籤 (label) 的置信度，以及修改至三張嬰兒相片（微笑、哭泣、看到傻臉而大笑）來應對不同情況。



躲貓貓：第二部分

延展你的專案

❑ 以下是一個更加進取的延展專案：為老師製作一個使用 Personal Image Classifier (PIC) 來自動簽到的應用程式。

- 在 PIC 系統中，為班級中的每個學生創建標籤（例如 “Eva”、“Phillip”、“Andrea” 等），並為每個標籤（即學生）從不同角度拍攝該學生的25張影像。

- 以這些影像來訓練模型。由於影像繁多，可能訓練需時。
- 然後測試你的模型，看看它可否正確識別隨機選中的學生。
- 如果模型的分類準確，而你又已構建好相應的應用程式，那麼老師可在課堂開始時，讓每個學生經過該開啟了程式的裝置。當學生以高置信度被識別時，便會被標記為「出席」。
- 你也可先以少量學生，製作一個簡化版的應用程式，從而理解指令方塊如何修改。
- 過程中，你可能會發現 PIC 系統中的某些先天限制。記下這些內容，並與老師分享，甚至讓 MIT 團隊知道。

❑ 你又有哪些新奇的主意呢？

躲貓貓：第二部分

運算思維概念

以下是我們在第二教節學會的運算思維概念。

L3UAI2 PICaboo
1. 事件
2. 條件
3. 數據結構

躲貓貓：第二部分

運算思維實踐

以下是我們在本單元學會的運算思維實踐。

L3UAI2 PICaboo
1. 反覆構思及漸進編程 首先訓練影像分類模型，然後將模型上傳到應用程式，再從分類結果中提取和使用信息來執行應用程式；最終通過一系列課堂內容，漸進構建複雜的應用程式。
2. 測試及除錯 測試 Personal Image Classifier (PIC) 有否根據瀏覽器上的影像訓練，而為影像準確分類；並且根據影像分類的置信度，測試流動裝置上的應用程式有沒產生正確的反應。
3. 算法思維 完成一種算法，使之能以圖形方式顯示置信度，並根據機器對這些置信度的反應作出決策。