

進制轉換器

作者：葉富豪

進制轉換器

本單元是透過進制轉換器實作，讓同學了解 APP Inventor 的應用，並學習到相關的程式邏輯概念。在 APP 使用介面，使用者輸入十進制的數字，經由 APP 轉換，將十進制的數字分別轉換成二進制、八進制、十六進制，並且將答案顯示在 APP 使用介面，總結以上我們將 APP 的需求整理成列表如下：

	需求名稱	需求說明	對應元件
需求 1	顯示數字	顯示各進制的數字	Label
需求 2	輸入數字	讓使用者輸入的想要轉換的數字	TextBox
需求 3	轉換	讓使用者轉換輸入的數字	Button
需求 4	警告	判斷使用者輸入的內容是否為數字，並給予警告	Notifier

進制轉換方式說明:

此單元內進制包含二進制、八進制、十進制、十六進制。

二進制:

二進制進位方式為，以 2 為基數且逢 2 進位。

例如：十進制的 1，在二進制為 1。

十進制的 2，在二進制為 10。

十進制的 3，在二進制為 11。

十進制的 4，在二進制為 100。

如個位數為 2，則個位數歸零，將十位數加一，以此類推……，所以二進制的數字符號只有 0 與 1。

十進制轉二進制計算方式:

十進制的 10 = 二進制的 1010

$10 / 2 = 5$	， 餘數為 0	↑ 由下往上，合併餘數則答案為 1010
$5 / 2 = 2$	， 餘數為 1	
$2 / 2 = 1$	， 餘數為 0	
$1 / 2 = 0$	， 餘數為 1	

八進制:

八進制進位方式為，以 8 為基數且逢 8 進位。

例如：十進制的 6，在八進制為 6。

十進制的 7，在八進制為 7。

十進制的 8，在八進制為 10。

十進制的 14，在八進制為 16。

十進制的 16，在八進制為 20。

如個位數為 8，則個位數歸零，將十位數加一，以此類推……，所以八進制的數字符號有 0、1、2、3、4、5、6、7。

十進制轉八進制計算方式:

十進制的 278 = 八進制的 426

$$\begin{array}{r} 278 / 8 = 34, \text{ 餘數為 } 6 \\ 34 / 8 = 4, \text{ 餘數為 } 2 \\ 4 / 8 = 0, \text{ 餘數為 } 4 \end{array} \quad \left. \begin{array}{l} \uparrow \\ \uparrow \\ \uparrow \end{array} \right\} \text{ 由下往上, 合併餘數則答案為 } 426$$

十六進制:

十六進制進位方式為，以 16 為基數且逢 16 進位。

例如：十進制的 6，在十六進制為 6。

十進制的 9，在十六進制為 9。

十進制的 10，在十六進制為 A。

十進制的 30，在十六進制為 1E。

十進制的 32，在十六進制為 20。

如個位數為 16，則個位數歸零，將十位數加一，以此類推……。

十六進制比較特別，因為個位數不能突破 10，所以從 10 開始以英文字母代替，10 等於 A、11 等於 B、12 等於 C、13 等於 D、14 等於 E、15 等於 F，16 則進位。所以十六進制的數字符號有 0、1、2、3、4、5、6、7、8、9。

A、B、C、D、E、F。

十進制轉十六進制計算方式:

十進制的 1456 = 十六進制的 5B0

$$\begin{array}{r} 1456 / 16 = 91, \text{ 餘數為 } 0 \\ 91 / 16 = 5, \text{ 餘數為 } 11 = B \\ 5 / 16 = 0, \text{ 餘數為 } 5 \end{array} \quad \left. \begin{array}{l} \uparrow \\ \uparrow \\ \uparrow \end{array} \right\} \text{ 由下往上, 合併餘數則答案為 } 5B0$$

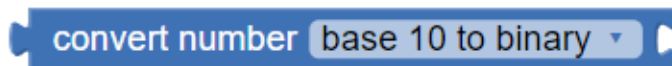
本單元將進制轉換分為三個部分說明，分別為 APP Inventor 內擁有的進制轉換，自製進制轉換器，以及將自製進制轉換器的程式帶入函式。

第一部分：APP Inventor 內建的十進制轉換器

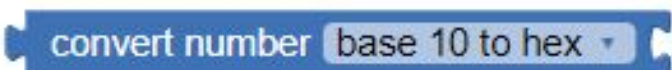
下圖為進制轉換器主要畫面：



第一部分將使用 APP Inventor 內建的進制轉換器，此進制轉換的功能在 Math 選單中，最下方名為 convert number 的程式方塊，如下圖：



上圖為十進制轉二進制。



上圖為十進制轉十六進制。

APP 程式完整圖:

```

initialize global Decimal to 0
initialize global Binary to 0
initialize global Hexadecimal to 0

when Conversion .Click
do
  set global Decimal to Decimal_content .Text
  if is number? get global Decimal
  then
    set BinaryAnswer .Text to convert number base 10 to binary get global Decimal
    set HexadecimalAnswer .Text to convert number base 10 to hex get global Decimal
  else
    call Notifier1 .ShowMessageDialog
      message "輸入錯誤!"
      title "訊息提示"
      buttonText "確定"
  
```

程式初始時，宣告十進制(Decimal)，二進制(Binary)，十六進制(Hexadecimal)數值為 0 的變數。

```

initialize global Decimal to 0
initialize global Binary to 0
initialize global Hexadecimal to 0
  
```

當轉換按鈕被點選時，將十進制的變數設定為文字輸入盒的內容，並且判斷輸入的內容是否為數字，若不是則以對話框顯示訊息對話框，訊息為「輸入錯誤!」，按鈕文字為「確定」。

若輸入內容為數字時，使用 APP Inventor 內擁有的 convert number，將十進制轉換為二進制與十六進制，並將結果顯示在二進制與十六進制的答案。

```

when Conversion .Click
do
  set global Decimal to Decimal_content .Text
  if is number? get global Decimal
  then
    set BinaryAnswer .Text to convert number base 10 to binary get global Decimal
    set HexadecimalAnswer .Text to convert number base 10 to hex get global Decimal
  else
    call Notifier1 .ShowMessageDialog
      message "輸入錯誤!"
      title "訊息提示"
      buttonText "確定"
  
```

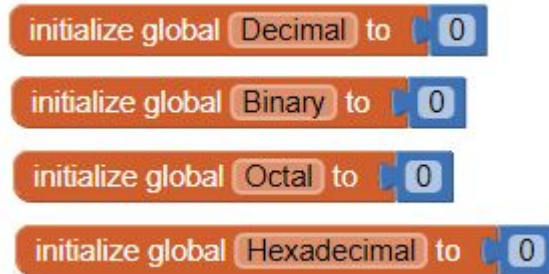
第二部分：自製十進制轉換器

下圖為進制轉換器主要畫面：

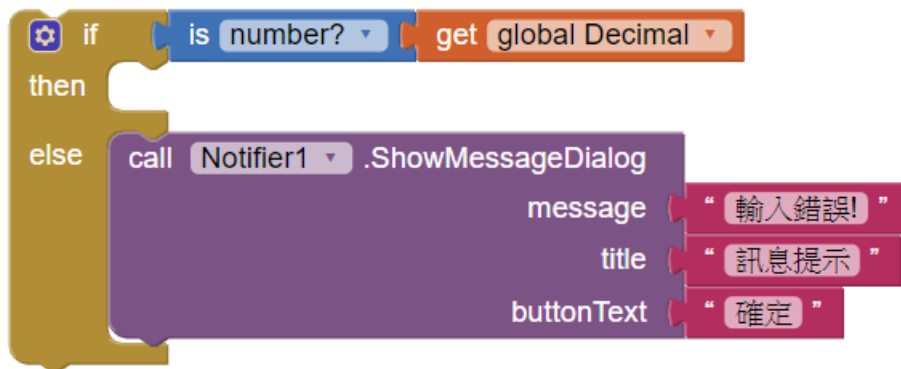


由於 APP Inventor 內的 convert number 只有十進制轉成二進制與十六進制，或是二進制與十六進制轉回十進制的功能，因此第二部分將自行設計十進制轉二進制、八進制以及十六進制功能的進制轉換器。

程式初始時，宣告十進制(Decimal)，二進制(Binary)，八進制(Octal)，十六進制(Hexadecimal)數值為 0 的變數。

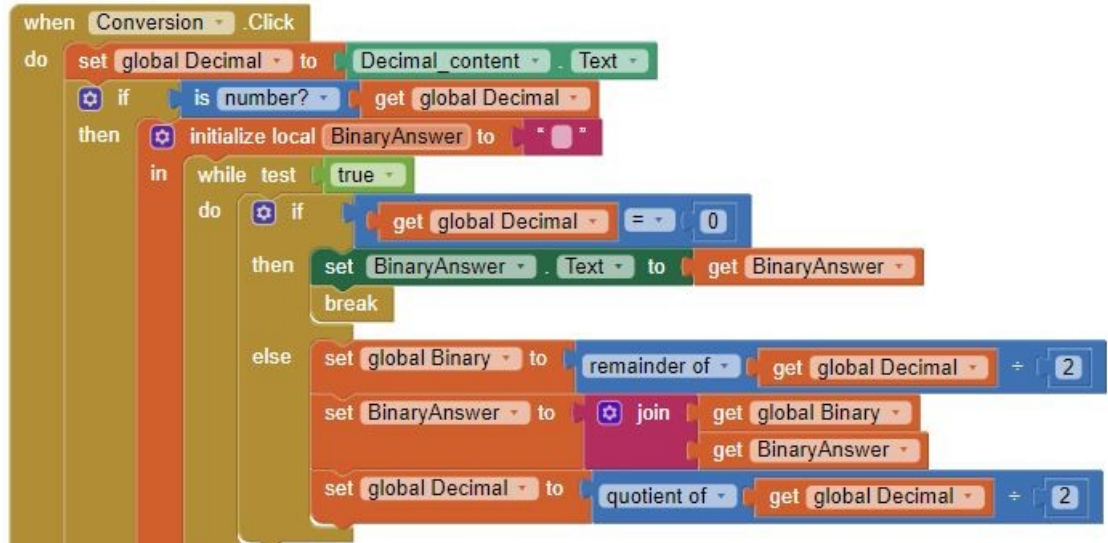


當轉換按鈕被點選時，將十進制的變數設定為文字輸入盒的內容，並且判斷輸入的內容是否為數字，若不是則以對話框顯示訊息對話框，訊息為「輸入錯誤!」，按鈕文字為「確定」。



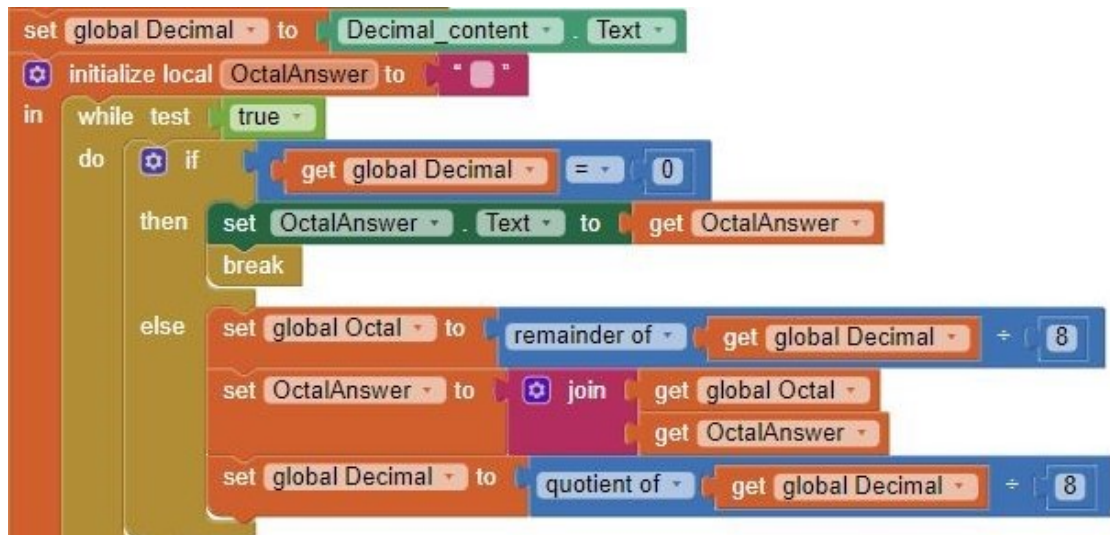
若輸入內容為數字時，則宣告字串為空的新變數 BinaryAnswer，以記錄二進制轉換的答案，利用 do while 迴圈，當條件為 true 時，執行迴圈；迴圈內，十進制不等於 0 時，十進制除以 2，設定十進制取其商數，二進制取其餘數，並將餘數合併至 BinaryAnswer，持續至十進制等於 0。

若十進制等於 0，將結果顯示在二進制的答案後，使用 break 終止迴圈。



根據二進制類推，將十進制的變數設定為文字輸入盒的內容，宣告字串為空的新變數 OctalAnswer，以記錄八進制轉換的答案，利用 do while 迴圈，當條件為 true 時，執行迴圈；迴圈內，十進制不等於 0 時，十進制除以 8，設定十進制取其商數，八進制取其餘數，並將餘數合併至 OctalAnswer，持續至十進制等於 0。

若十進制等於 0，將結果顯示在八進制的答案後，使用 break 終止迴圈。



根據二進制類推，將十進制的變數設定為文字輸入盒的內容，宣告字串為空的新變數 HexadecimalAnswer，以記錄十六進制轉換的答案，利用 do while 迴圈，當條件為 true 時，執行迴圈；迴圈內，十進制不等於 0 時，十進制除以 16，設定十進制取其商數，十六進制取其餘數，若餘數超越 10 時，使用對應的英文字母代替數字。

- 例如：
- 餘數為 10 時，餘數轉換為 A。
 - 餘數為 11 時，餘數轉換為 B。
 - 餘數為 12 時，餘數轉換為 C。
 - 餘數為 13 時，餘數轉換為 D。
 - 餘數為 14 時，餘數轉換為 E。
 - 餘數為 15 時，餘數轉換為 F。

完成後，將餘數合併至 HexadecimalAnswer，持續至十進制等於 0。

若十進制等於 0，將結果顯示在十六進制的答案後，使用 break 終止迴圈。

```

set global Decimal to Decimal_content.Text
initialize local HexadecimalAnswer to ""
in while test true
do
  if get global Decimal = 0
  then
    set HexadecimalAnswer.Text to get HexadecimalAnswer
    break
  else
    set global Hexadecimal to remainder of get global Decimal ÷ 16
    if get global Hexadecimal = 10
    then
      set global Hexadecimal to join " A "
    else if get global Hexadecimal = 11
    then
      set global Hexadecimal to join " B "
    else if get global Hexadecimal = 12
    then
      set global Hexadecimal to join " C "
    else if get global Hexadecimal = 13
    then
      set global Hexadecimal to join " D "
    else if get global Hexadecimal = 14
    then
      set global Hexadecimal to join " E "
    else if get global Hexadecimal = 15
    then
      set global Hexadecimal to join " F "
    set HexadecimalAnswer to join get global Hexadecimal
      get HexadecimalAnswer
    set global Decimal to quotient of get global Decimal ÷ 16
  
```


APP 程式完整圖:

```

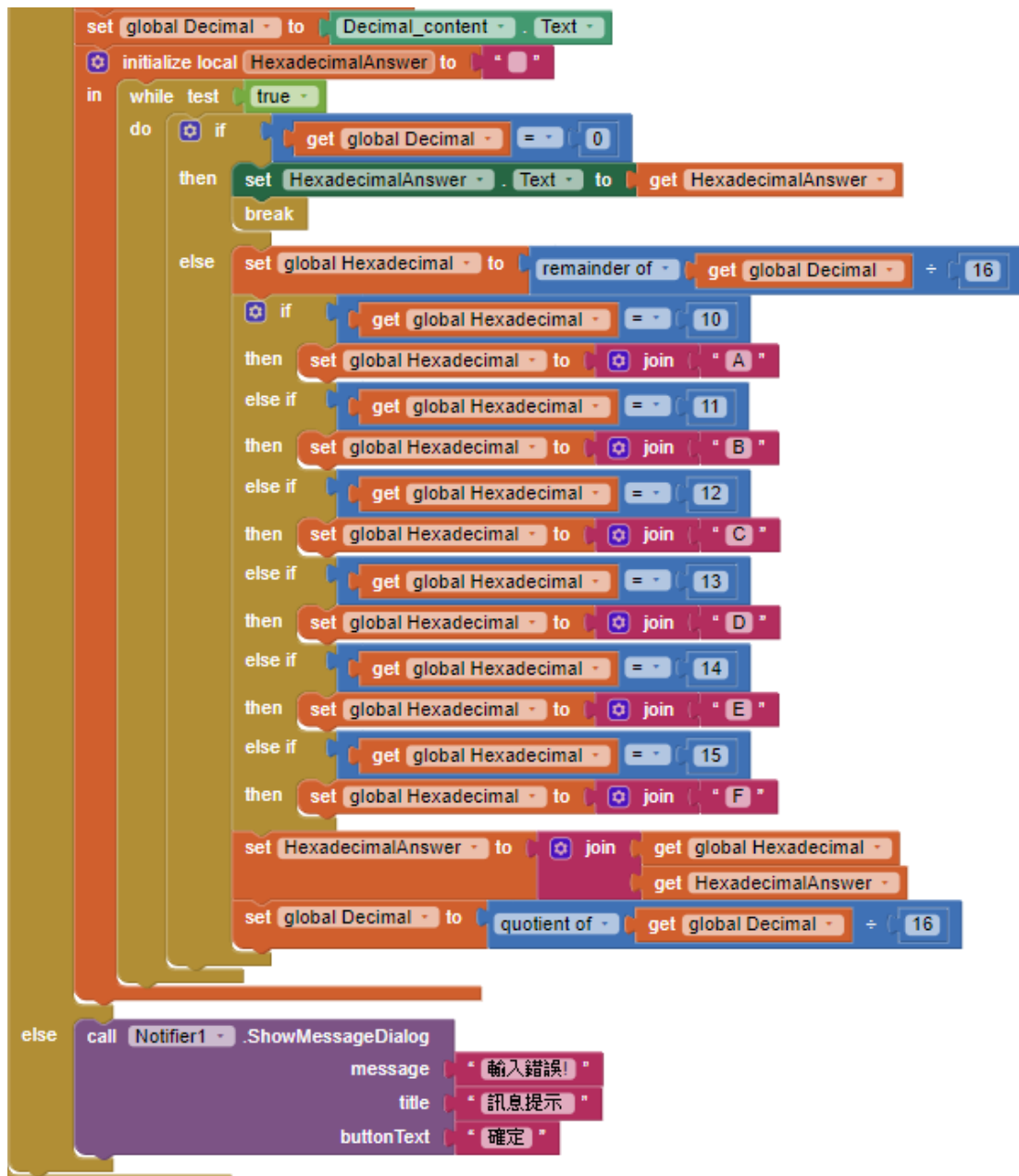
initialize global Decimal to 0
initialize global Binary to 0
initialize global Octal to 0
initialize global Hexadecimal to 0

```

```

when Conversion .Click
do
  set global Decimal to Decimal_content .Text
  if is number? get global Decimal
  then
    initialize local BinaryAnswer to ""
    in while test true
    do
      if get global Decimal = 0
      then
        set BinaryAnswer .Text to get BinaryAnswer
        break
      else
        set global Binary to remainder of get global Decimal ÷ 2
        set BinaryAnswer to join get global Binary
        get BinaryAnswer
        set global Decimal to quotient of get global Decimal ÷ 2
      end if
    end do
  end if
  set global Decimal to Decimal_content .Text
  initialize local OctalAnswer to ""
  in while test true
  do
    if get global Decimal = 0
    then
      set OctalAnswer .Text to get OctalAnswer
      break
    else
      set global Octal to remainder of get global Decimal ÷ 8
      set OctalAnswer to join get global Octal
        get OctalAnswer
      set global Decimal to quotient of get global Decimal ÷ 8
    end if
  end do
end do

```



第三部分：自製十進制轉換器(函式)

因第二部分的程式太過於龐大以及冗長，因此第三部分將使用函式來縮短程式，避免日後程式相同時，需要重新編寫程式，只需呼叫函式即可。

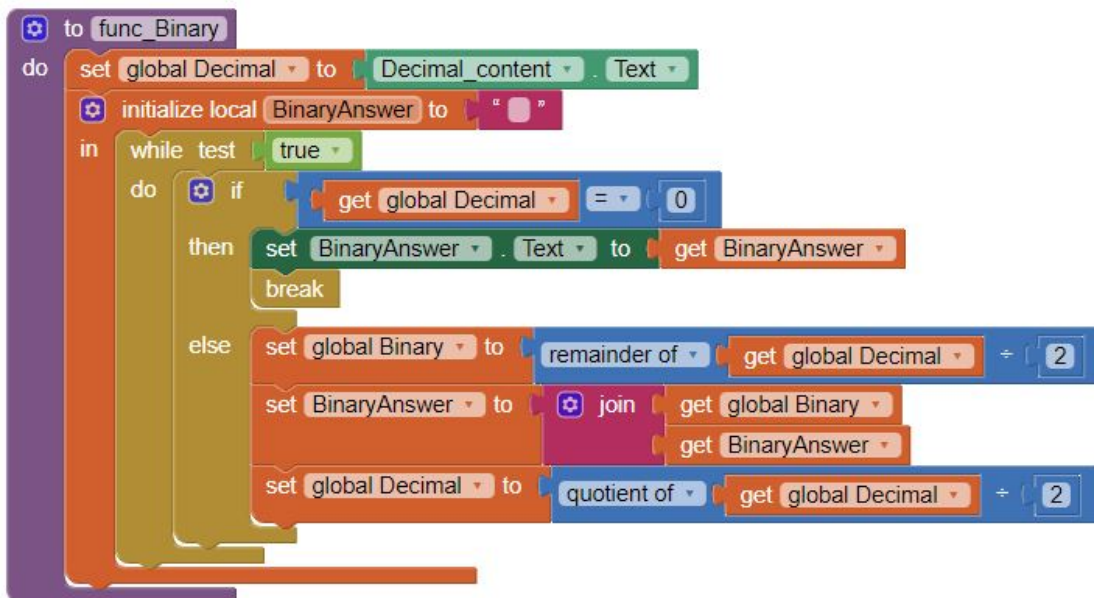
根據上述述說，第三部分的修改主要為程式，因此進制轉換器的主要畫面不需更改。

程式初始時，宣告十進制(Decimal)，二進制(Binary)，八進制(Octal)，十六進制(Hexadecimal)數值為 0 的變數。



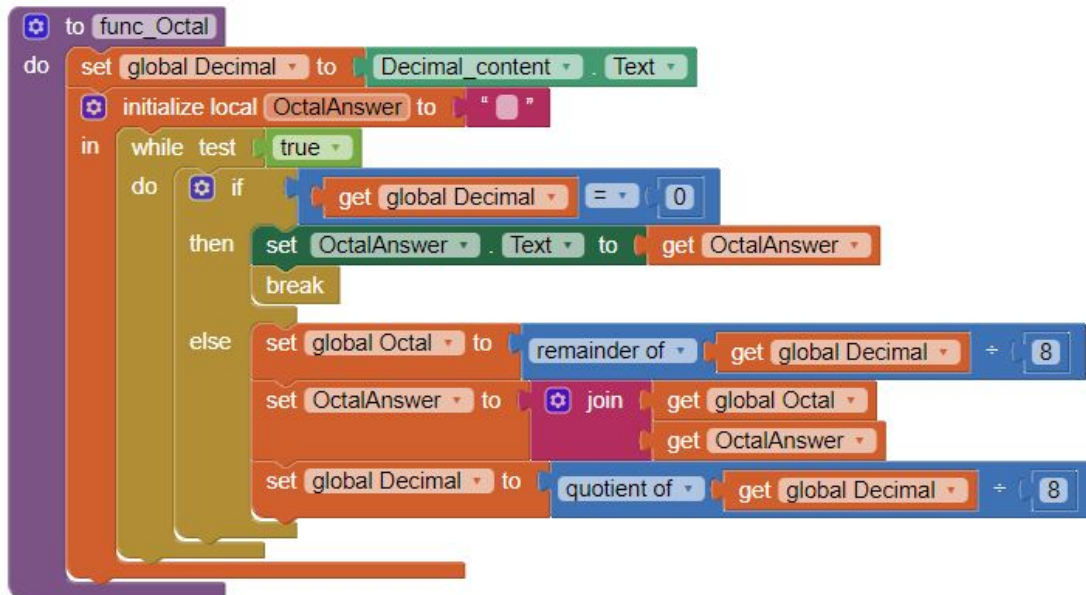
宣告名為 func_Binary 的函式，將十進制的變數設定為文字輸入盒的內容，則宣告字串為空的新變數 BinaryAnswer，以記錄二進制轉換的答案，利用 do while 迴圈，當條件為 true 時，執行迴圈；迴圈內，十進制不等於 0 時，十進制除以 2，設定十進制取其商數，二進制取其餘數，並將餘數合併至 BinaryAnswer，持續至十進制等於 0。

若十進制等於 0，將結果顯示在二進制的答案後，使用 break 終止迴圈。



宣告名為 `func_Octal` 的函式，將十進制的變數設定為文字輸入盒的內容，則宣告字串為空的新變數 `OctalAnswer`，以記錄八進制轉換的答案，利用 `do while` 迴圈，當條件為 `true` 時，執行迴圈；迴圈內，十進制不等於 0 時，十進制除以 8，設定十進制取其商數，八進制取其餘數，並將餘數合併至 `OctalAnswer`，持續至十進制等於 0。

若十進制等於 0，將結果顯示在八進制的答案後，使用 `break` 終止迴圈。



```
to func_Octal
do
  set global Decimal to Decimal_content . Text
  initialize local OctalAnswer to ""
  in while test true
  do
    if get global Decimal = 0
    then
      set OctalAnswer . Text to get OctalAnswer
      break
    else
      set global Octal to remainder of get global Decimal ÷ 8
      set OctalAnswer to join get global Octal
      get OctalAnswer
      set global Decimal to quotient of get global Decimal ÷ 8
```


宣告名為 `func_Hexadecimal` 的函式，將十進制的變數設定為文字輸入盒的內容，則宣告字串為空的新變數 `HexadecimalAnswer`，以記錄十六進制轉換的答案，利用 `do while` 迴圈，當條件為 `true` 時，執行迴圈；迴圈內，十進制不等於 0 時，十進制除以 16，設定十進制取其商數，十六進制取其餘數，根據餘數轉換成相對應的英文字母，並將餘數合併至 `HexadecimalAnswer`，持續至十進制等於 0。

若十進制等於 0，將結果顯示在十六進制的答案後，使用 `break` 終止迴圈。

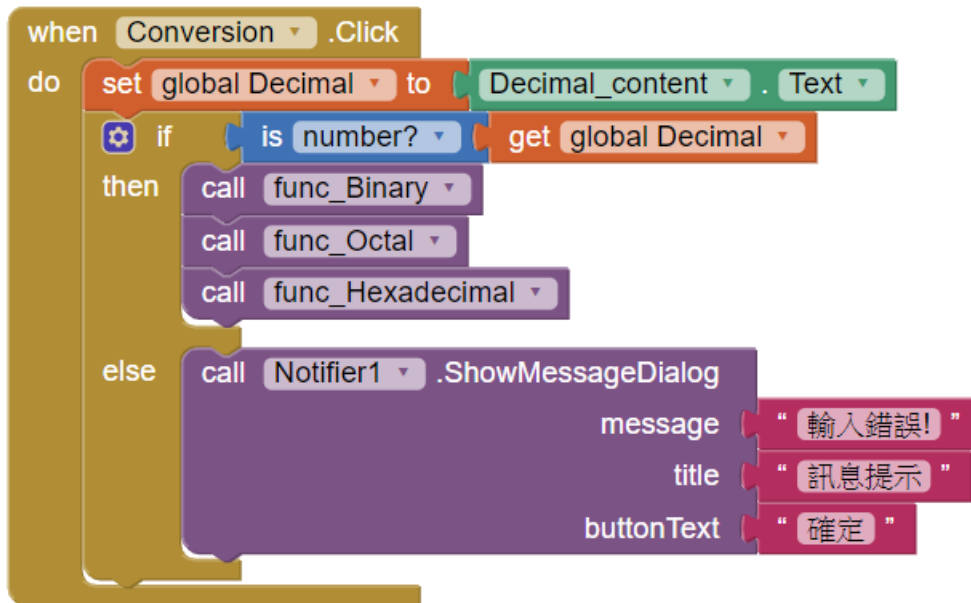
```

to func_Hexadecimal
do
  set global Decimal to Decimal_content.Text
  initialize local HexadecimalAnswer to ""
  in while test true
  do
    if get global Decimal = 0
    then
      set HexadecimalAnswer.Text to get HexadecimalAnswer
      break
    else
      set global Hexadecimal to remainder of get global Decimal ÷ 16
      if get global Hexadecimal = 10
      then set global Hexadecimal to join "A"
      else if get global Hexadecimal = 11
      then set global Hexadecimal to join "B"
      else if get global Hexadecimal = 12
      then set global Hexadecimal to join "C"
      else if get global Hexadecimal = 13
      then set global Hexadecimal to join "D"
      else if get global Hexadecimal = 14
      then set global Hexadecimal to join "E"
      else if get global Hexadecimal = 15
      then set global Hexadecimal to join "F"
      set HexadecimalAnswer to join get global Hexadecimal
      get HexadecimalAnswer
      set global Decimal to quotient of get global Decimal ÷ 16
  end
end

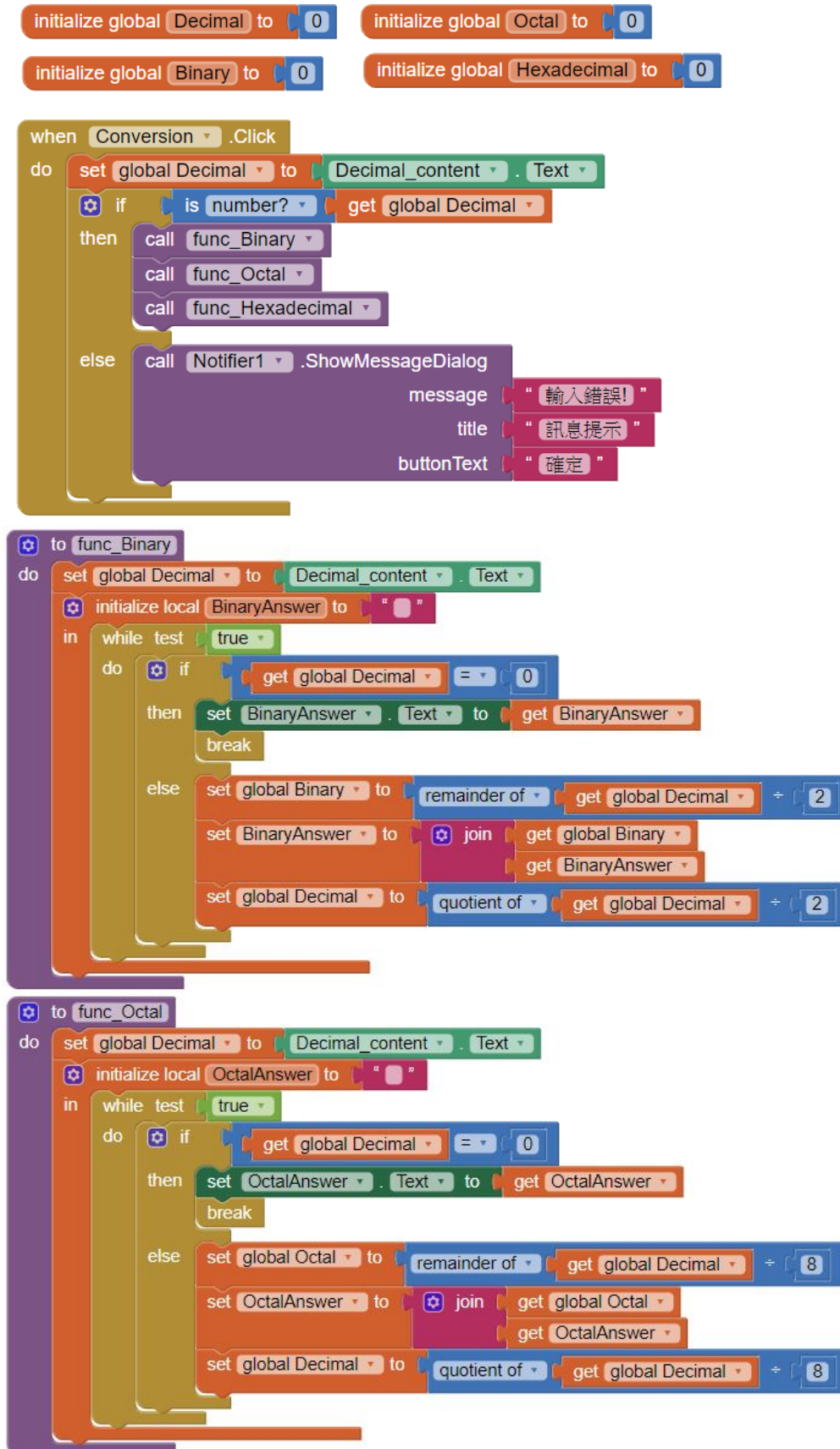
```

當轉換按鈕被點選時，將十進制的變數設定為文字輸入盒的內容，並且判斷輸入的內容是否為數字，若不是則以對話框顯示訊息對話框，訊息為「輸入錯誤!」，按鈕文字為「確定」。

若輸入內容為數字時，呼叫二進制、八進制以及十六進制的函式，來轉換十進制。



APP 程式完整圖:



```

to func_Hexadecimal
do
  set global Decimal to Decimal_content . Text
  initialize local HexadecimalAnswer to ""
  in while test true
  do
    if get global Decimal = 0
    then
      set HexadecimalAnswer . Text to get HexadecimalAnswer
      break
    else
      set global Hexadecimal to remainder of get global Decimal ÷ 16
      if get global Hexadecimal = 10
      then set global Hexadecimal to join " A "
      else if get global Hexadecimal = 11
      then set global Hexadecimal to join " B "
      else if get global Hexadecimal = 12
      then set global Hexadecimal to join " C "
      else if get global Hexadecimal = 13
      then set global Hexadecimal to join " D "
      else if get global Hexadecimal = 14
      then set global Hexadecimal to join " E "
      else if get global Hexadecimal = 15
      then set global Hexadecimal to join " F "
      set HexadecimalAnswer to join get global Hexadecimal
      get HexadecimalAnswer
      set global Decimal to quotient of get global Decimal ÷ 16
    end if
  end do
end do

```

範例:

進制轉換器

十進制: 123 轉換

二進制:

1111011

八進制:

173

十六進制:

7B