

APPENDIX A

{ App Inventor 指令集 : BuiltIn }

A-1 Definition 指令區	408
A-2 Text 文字指令區	410
A-3 List 清單指令區	413
A-4 Math 數學指令區	416
A-5 Logic 邏輯指令區	421
A-6 Control 控制指令區	422
A-7 Color 顏色指令區	424



A-1 Definition 指令區

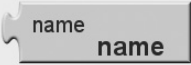


procedure

procedureWithResult

name

variable

中文名稱	英文名稱	圖形
副程式	procedure	
功能	<p>將多個指令集合在一起，之後可透過呼叫該副程式來使用這些指令。如果該副程式包含了參數，則必須使用 name 指令來命名該參數。當建立一個副程式之後，App Inventor 會自動產生一個呼叫（call）指令，位於 My Definitions 之中，您可使用該指令來呼叫對應的副程式。</p> <p>當建立一個新的副程式時，App Inventor 會自動幫它取一個名稱，您也可以點選它之後自行改成您所需要的名稱。在一個程式中的副程式名稱必須是唯一的，App Inventor 不允許在同一個程式中有兩個名稱相同的副程式。您可點選程序上的標籤來將其重新命名。App Inventor 會自動調整對應的呼叫指令名稱</p>	
具回傳值的副程式	procedureWithResult	
功能	<p>本指令與副程式相同，但使用時會回傳一個結果（return）。當本程序執行完畢後會將 return 欄位回傳。</p>	

名稱	name	
功能	<p>您可建立一個 <code>name</code> 參數來呼叫指定的程序。定義參數的方法就是將 <code>name</code> 插入 <code>procedure</code> 的 <code>arg</code> 插槽。可使用的參數數目不限，每當您填入一個參數插槽時，就會自動跳出新的參數插槽。只要點選右下方的 <code>name</code> 標籤就可將其重新命名。</p> <p>當建立一個副程式之後，App Inventor 會自動產生一個呼叫（<code>call</code>）指令，位於 My Definitions 之中，您可使用該指令來呼叫對應的副程式。</p> <p>App Inventor 會自動幫您所建立的名稱參數建立對應的取值指令並將它們放在 My Definitions 中。您可使用這些指令來存取該參數的值，當該程序被呼叫時，該值就會被傳送過去給程序。</p> <p>注意：現在的 App Inventor 版本規定一個程式中的所有參數名稱都必須是唯一的，即便是在不同副程式下的參數也是一樣（C、Java 就無此限制）。例如兩個不同程序就不能都使用名為 <code>x</code> 的變數。當您要建立第二個 <code>x</code> 時，App Inventor 就無法這麼做。</p>	
變數	variable	
功能	<p>建立一個在程式執行時可以動態改變的值，別忘了給它一個名字。在此所建立的變數為全域變數（<code>global variable</code>），代表程式中的任何一個指令都可以存取此變數，當然也可以從 <code>procedure</code> 中來呼叫。</p> <p>當建立一個新的變數時，App Inventor 會自動幫它取一個名稱，您也可以點選它之後自行改成您所需要的名稱。在一個程式中的變數名稱必須是唯一的，App Inventor 不允許在同一個程式中有兩個名稱相同的變數。</p> <p>另一方面在建立變數完成後，App Inventor 會自動產生兩個相關的指令，它們位於 My Definitions drawer 中：</p> <ul style="list-style-type: none"> <code>global</code> 指令用來取得變數值。 <code>set global</code> 指令用來設定變數值。 <p>您在編寫程式的過程中隨時可以修改變數的名稱，修改完成之後對應的指令也會自動修正名稱。</p>	
轉接頭		



功能

本指令是為了左側沒有對應插槽的指令所產生的，本指令就好像轉接頭一樣，並沒有程式上的作用。這時候您轉接的指令就可以執行了，但它們如果有回傳值的話在此會被忽略。有時候當我們建立一個具回傳值的副程式，但想要在背景呼叫它時（此時不能使用回傳值），這個功能就很好用了。



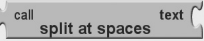
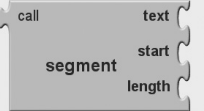
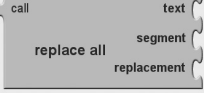
A-2 Text 文字指令區

text	downcase
=	starts at
join	contains
make text	split at first
length	split at first of any
text<	split
text=	split at any
text>	split at spaces
trim	segment
upcase	replace all

名稱	圖形	功能
text		指定一個字串常數。
join		將兩個指定字串合成一個新字串。對於此指令，數字也被視為字串。例如：用 join 指令來合成 1+1 及 2*3 則回傳 26 (2 和 6 合成)。
make text		將所有的內容依序合成一個字串。
length		回傳指定字串的長度。(整數)
text<		回傳第一個字串 text1 是否在字母排列上比第二個字串 text2 小。若兩者的第一個字母相同則比較字串長度。大寫字母比小寫字母優先。(布林)
text=		回傳第一個字串 text1 與第二個字串 text2 是否相等。請注意如果要比較的字串中含有數字，則數學上的 = 與字串上的 text= 結果是不一樣的。如果兩個 textbox，其中一個內容為 123，另一個是 0123，則數學上比較結果為相等，但字串上比較則不相等。(布林)
text>		回傳第一個字串 text1 是否在字母排列上比第二個字串 text2 大。若兩者的第一個字母相同則比較字串長度。大寫字母比小寫字母優先。(布林)
upcase		將指定字串全部轉為大寫。
downcase		將指定字串全部轉為小寫。



trim		刪除指定字串的頭尾空格。
starts at		<p>回傳指定子字串在指定字串中的位置，找不到則回傳 0。例如子字串「ana」在「Havana」中的位置為 4。</p> <p>請注意！在一般程式設計語言中，陣列的第一個元素編號為 0，但 AppInventor 中的第一個元素編號為 1。</p>
contains		若指定子字串出現在指定字串中則回傳真，反之則回傳假。
split at first		將字串從指定分割點 (at) 第一次出現的地方分成兩個子字串，並回傳一個包含這兩個子字串的清單，一個是從原字串第一個字母到分割點前一個字母，另一個則是分割點後一個字母到原字串結尾。例如將字串「apple,banana,cherry,dogfood」使用逗號「,」來分割，回傳結果會是兩個子字串：第一個子字串為「apple」，第二個子字串為「banana,cherry,dogfood」。請注意逗號「,」這個分割點不包含在任何一個子字串中。(清單)
split at first of any		將字串從指定分割點 (at) 分割成兩個子字串，並回傳一個包含這兩個子字串的清單。(清單)
split		將字串從指定分割點切割，並以清單回傳切割結果。例如將字串「one,two,three,four」從逗號「,」分割的結果是 (one two three four) 這個清單。將字串「one-potato,two-potato,three-potato,four」從「at -potato」分割的結果是 (one two three four) 這個清單。(清單)
split at any		將字串從清單項目 (at) 來切割，意即使用清單項目來切割字串，並回傳結果。例如將字串「appleberry,banana,cherry,dogfoodwith」以一個具有兩個項目的清單來切割，第一個項目為逗號「,」，第二個項目為「rry」，切割結果為 (applebe banana che dogfood) 這個字串。(清單)

split at spaces		將指定字串在所有空格處分開，以清單輸出結果。 (清單)
segment		將原字串從指定位置 start 開始並指定長度 length 後產生子字串。(字串)
replace all		將原字串以新的子字串取代後回傳新的字串。(字串)

A-3 List 清單指令區

make a list

select list item

replace list item

remove list item

insert list item

length of list

append to list

add items to list

is in list?

position in list

pick random item

is list empty?

copy list

is a list?

list to csv row

list to csv table

list from csv row

list from csv table



App Inventor 中的 list 就是一般程式中的陣列，所以相關操作方法皆與陣列非常相似。但請注意 App Inventor 的 List 編號從 1 開始，陣列編號則從 0 開始。

名稱	圖形	功能
make a list		新增一個清單，並自行指定其元素 (item)。若您未指定任何元素，則此為一空清單，您可以之後再加入元素。
select list item		取得清單 list 的指定位置 index 元素內容，第一個清單元素位置為 1。
replace list item		將清單的指定位置元素以新的內容 replacement 取代。
remove list item		從清單中刪除指定位置的元素。
insert list item		將指定內容 item 插入清單的指定位置。
length of list		回傳清單的長度，也就是元素數目。(整數)
append to list		將第一個清單 list1 與第二個清單 list2 組成一個新的清單。
add items to list		將指定內容 item 接在清單的後面。本指令與 append to list 指令的差別在於 append to list 指令是將兩個清單組起來，而 add items to list 指令是將要新增的內容當作個別參數來操作，會自動產生新的插槽。

is in list?		若指定內容 thing 存在於清單中回傳 true ，反之回傳 false 。注意：若一清單中含有子清單，則子清單的元素不包含在原清單中。例如清單 (1 2 (3 4)) 的元素為 1、2 以及子清單 (3 4)；單獨 3 或 4 並非這個 list 的元素。
position in list		回傳指定內容 thing 位於清單中的位置，回傳 0 代表清單不包含指定內容。(整數)
pick random item		從清單中隨機取得任一項目。
is list empty?		如果清單為空，回傳 true ；反之回傳 false 。
copy list		複製清單，如果清單包含子清單也會一併複製。
is a list?		如果指定內容格式為清單，回傳 true ；反之回傳 false 。
list to csv row		將清單轉換為 CSV 表格中的列 row ，並以 CSV (comma-separated value) 格式回傳。Row 中的每一個項目就是一個欄位 (field)。回傳的文件結尾不會包含換行符號。
list to csv table		將清單以列優先的方式轉換為 CSV 表格，並以 CSV 格式回傳。回傳清單中的項目是另一個清單，代表 CSV 表格中的列，每列中的項目則是該列的欄位。列中的項目是以逗號分隔，列彼此之間則是以 CRLF (\r\n) 符號分隔。
list from csv row		將 CSV 文件中的列解析並回傳一個清單，清單內容就是該列的各個欄位。不同列將以 \n 或 CRLF (\r\n) 符號來區隔。如果列中的文字是以新的一列或是 CRLF 符號結束，這樣的語法是允許的。




list from csv table		<p>將 CSV 表格解析並回傳一個清單，清單內容代表不同的列（再包含不同的欄位）。不同列將以 \n 或 CRLF (\r\n) 符號來區隔。</p>
---------------------	---	---

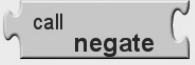
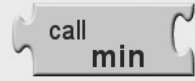

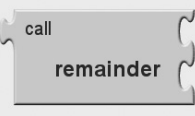
A-4 Math 數學指令區

- | | |
|-----------------|-------------------|
| number | modulo |
| > | abs |
| < | round |
| <= | floor |
| >= | ceiling |
| equals | expt |
| + | exp |
| - | log |
| x | sin |
| ÷ | cos |
| sqrt | tan |
| random fraction | asin |
| random integer | acos |
| random set seed | atan |
| negate | atan2 |
| min | format as decimal |
| max | is a number? |
| quotient | number |
| remainder | |

名稱	圖形	功能
number		指定一個數字常數。

>		比較兩個指定數字。如果前者大於後者時，回傳 <code>true</code> ，否則回傳 <code>false</code> 。
<		比較兩個指定數字。如果前者小於後者時，回傳 <code>true</code> ，否則回傳 <code>false</code> 。
<=		比較兩個指定數字。如果前者小於或等於後者時，回傳 <code>true</code> ，否則回傳 <code>false</code> 。
>=		比較兩個指定數字。如果前者大於或等於後者時，回傳 <code>true</code> ，否則回傳 <code>false</code> 。
equals		比較兩個指定數字是否相等。如果是的話，回傳 <code>true</code> ，否則回傳 <code>false</code> 。此指令和在 <code>logic</code> 指令區中的指令功能相同，僅為了方便所以兩邊都放。本指令的參數可為其他資料型態，而不僅僅是數字。
Not =?		比較兩個指定數字是否不相等。如果是的話，回傳 <code>true</code> ，否則回傳 <code>false</code> 。
+		回傳兩個指定數字的和。
-		回傳兩個指定數字的差。
×		回傳兩個指定數字的積。
÷		回傳前者除以後者的商。例如，1 除以 3 為 0.3333。



sqrt		回傳指定數字的平方根。
random fraction		回傳一個介於 0 和 1 之間的隨機小數。
random integer		回傳一個介於指定數字之間的隨機整數，包含上限 (to) 與下限 (from)。參數由小到或大到小不會影響計算結果。
negate		回傳指定數字的相反數。
min		回傳指定數字中最小者。(清單)
max		回傳指定數字中最大者。(清單)
quotient		quotient(a,b)指令回傳第一個數 a 除以第二個數 b 的商，但只取整數部分，小數點以後忽略不計。
remainder		remainder (a,b) 指令可回傳第一個數 a 除以第二個數 b 的餘數 (remainder)。例如 remainder (11,5) 為 1；remainder (-11,5) 為 -1；remainder (11,-5) 為 1；remainder (-11,-5) 為 -1。
modulo		當指定數字皆為正數時，Modulo (a,b) 計算結果與 remainder (a,b) 相同。一般公式如下：對所有 a 與 b 而言，(floor (a/b) × b) + modulo (a,b) = a。例如 modulo (11,5) 為 1；modulo (-11,5) 為 4；modulo (11,-5) 為 -4；modulo (-11,-5) 為 -1。Modulo (a,b) 永遠與 b 同號，但 remainder (a,b) 則永遠與 a 同號。

abs		回傳指定數字的絕對值。
round		回傳指定數字四捨五入到整數位的運算結果。
floor		回傳指定數字無條件捨去到整數位的運算結果。
ceiling		回傳指定數字無條件進位到整數位的運算結果。
expt		回傳指數的運算結果。(base : 底 exponent : 指數)。
exp		回傳 e (2.71828...) 的指定次方運算結果。
log		回傳指定數字的自然對數運算結果。
sin		回傳指定數字的正弦函數值，單位：度。
cos		回傳指定數字的餘弦函數值，單位：度。
tan		回傳指定數字的正切函數值，單位：度。



asin		回傳指定數字的反正弦函數值。
acos		回傳指定數字的反餘弦函數值。
atan		回傳指定數字的反正切函數值。
atan2		回傳 y/x 的反正切函數值。
convert radians to degrees		將弧度轉換為角度。
convert degrees to radians		將角度轉換為弧度。
format as decimal		將原數字轉換為指定位數之小數。指定小數位數 (places) 不能為負數。若原小數位數過多則四捨五入，反之則補 0。
is a number?		指定物件如果為數字，回傳 true，反之回傳 false。

A-5 Logic 邏輯指令區

true	=
false	and
not	or

名稱	圖形	功能
true		布林常數的真 (true)。用來設定元件的布林 (boolean) 屬性值，或用來表示某種狀況是否成立。
false		布林常數的假 (false)。用來設定元件的布林屬性值，或用來表示某種狀況是否不成立。
not		邏輯運算的 not。輸入 true 或條件判斷為 true 則回傳 false，反之回傳 true。
equals		綜合性的等於運算符。 兩個數字是否相等 (例如：1=1.0)。 兩個字串是否相等，包括大小寫。例如 banana 不等於 Banana。 若兩個清單的長度相同且對應元件相等則相等。
and		測試是否所有的敘述皆為真。當您插入第一個 test 項目時會自動跑出第二個插槽。測試順序由上到下，測試過程中若任一條件已知為假則停止測試。若無任何敘述則回傳 true。
or		測試所有敘述中是否至少有一者為真。當您插入第一個 test 項目時會自動跑出第二個插槽。測試順序由上到下，測試過程中若任一條件已知為真則停止測試。若無任何敘述則回傳 false。



A-6 Control 控制指令區

if

if...else

choose

foreach

for range

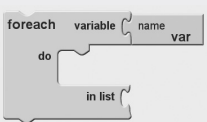
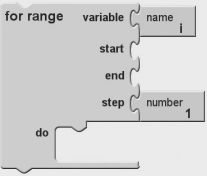
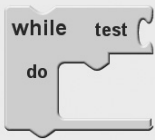

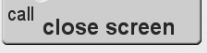
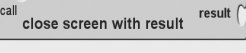
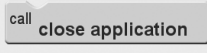
while

get start text

close screen

close screen with result

名稱	圖形	功能
if		測試指定條件 test。若為 true 則執行 then-do 區段中的動作，反之則跳過此段。
if...else		測試指定條件 test。若為 true 則執行 then-do 以下區塊，反之則則執行 else-do 以下區塊。
choose		測試指定敘述。若為 true 則執行 then-do 以下區塊並回傳 then-return 之值，反之則執行 else-do 以下區塊並回傳 else-return 之值。

foreach	 The 'foreach' block has a 'do' sub-block and an 'in list' input. The 'do' sub-block has a 'variable' input and a 'name' output labeled 'var'.	根據指定清單的元素數目來決定 do 的執行次數。您可使用該變數名稱 (var) 來取得它的值。
for range	 The 'for range' block has a 'do' sub-block. The 'do' sub-block has a 'variable' input and a 'name' output labeled 'i'. The 'for range' block has three inputs: 'start', 'end', and 'step', and one output labeled 'number' with a '1' below it.	根據指定範圍之整數個數來決定 do 的執行次數，可自由設定每次累加的數字 step。您可使用該變數名稱 (i) 來取得它的值。
while	 The 'while' block has a 'test' input and a 'do' sub-block.	測試指定條件 test。若為 true 則重複執行以下動作，反之則結束此段。
get start text	 The 'get start text' block has a 'call' input and a 'get start text' output.	如果有本項設定，當程式啟動時回傳所接收到的字串資訊。
close screen	 The 'close screen' block has a 'call' input and a 'close screen' output.	關閉螢幕。
close screen with result	 The 'close screen with result' block has a 'call' input and a 'close screen with result' output, with a 'result' output label.	關閉螢幕，並指定回傳結果 result。
close application	 The 'close application' block has a 'call' input and a 'close application' output.	結束程式。



A-7 Color 顏色指令區

顏色指令區中包含了各種常用的顏色指令，您可將它們直接插入需要顏色設定的指令中，例如設定文字顏色或是某元件的背景顏色。這些顏色也可以在 Designer 頁面中的相關屬性中來設定。

App Inventor 中內建的顏色有無色、黑色、藍色、青色 (cyan)、深灰色、灰色、綠色、淺灰色、洋紅色 (magenta)、橘色、粉紅色、紅色、白色與黃色。

如何自訂顏色

App Inventor 中的顏色是以數字進行編碼，例如藍色是由 -16776961 這個負數所代表。如果您知道如何自定顏色色碼，就不必受限於 App Inventor 那少少的內建顏色了，可以在您的 app 中自由創造各種色彩。

顏色是由 4 個參數所共同決定的，每個參數的範圍都是 0 ~ 255。前三個參數代表紅、綠、藍三原色的強度。您可以從網路上找到各種顏色的 RGB 色表，當然也可以自己實驗一下來找到喜歡的顏色。第 4 個數字則是「不透明度 (opacity)」，代表多個元件疊在一起時的視覺效果。Opacity 255 是完全不透明，該顏色會將其下層所有的東西都蓋過去。Opacity 0 則是完全透明，您在螢幕上甚至看不到這個顏色了。介於兩者中間的數值則決定該顏色底下東西的可視程度。

依照下列步驟要求出某種顏色在 App Inventor 中的數字代碼：

1. 先設定不透明度 (opacity)。
2. 乘以 256 之後加上 R (紅色強度)。
3. 第 2 步計算結果乘以 256 之後加上 G (綠色強度)。
4. 第 3 步計算結果乘以 256 之後加上 B (藍色強度)。
5. 第 4 步計算結果減掉 168 (4294967296)。
6. 計算結果 (皆為負數) 就是用在 App Inventor 中的顏色代碼。

請看以下範例，深紫色的 RGB 設定為 R=160，G=32，B=240。請將這些數值代入以上公式，並將 opacity 設為 255，計算結果為 -6283024。如果您將畫布的背景顏色設為這個數值，就可以看到螢幕變成漂亮的紫色了。本範例是藉由一個副程式（colorNumber）來指定 RGB 值來得到特定的顏色，不透明度都固定為 255，您也可以自由調整不透明度的設定值。



圖 A-1 自訂顏色範例

