

# APPENDI X B

## { App Inventor 指令集 : MyBlocks }

B-1 Basic 基本元件	428
B-2 Media Component 多媒體元件	454
B-3 Animation 動畫元件	462
B-4 Social Components 通訊元件	471
B-5 Sensor 感應器元件	489
B-6 Screen Arrangement 螢幕排列元件	495
B-7 LEGO MINDSTORMS 樂高機器人控制元件	499
B-8 Other Components 其他元件	520



# B-1 Basic 基本元件

- |          |                 |
|----------|-----------------|
| Button   | ListPicker      |
| Canvas   | PasswordTextBox |
| CheckBox | Screen          |
| Clock    | TextBox         |
| Image    | TinyDB          |
| Label    |                 |

Button 按鈕	圖形	
	功能	
	<p>按鈕元件可在程式中設定特定的觸碰動作。</p> <p>按鈕可知道使用者是否正在按它。您可自由調整按鈕的各種外觀屬性，或使用 Enabled 屬性決定按鈕是否可以被點擊。</p>	
	屬性	
	<p><b>BackgroundColor</b> 文字背景顏色。</p> <p>set <b>Button1.BackgroundColor</b> to <b>Button1.BackgroundColor</b></p>	
<p><b>Enabled</b> 如果本項屬性設定為真，則按鈕可被點選，反之則無法點選。</p> <p>set <b>Button1.Enabled</b> to <b>Button1.Enabled</b></p>		
<p><b>FontBold</b> 文字為粗體。</p>		
<p><b>FontItalic</b> 文字為斜體。</p>		

Button  
按鈕**FontSize**

文字字體大小。

**FontTypeFace**

按鈕文字的字體種類。

**Height**

按鈕高度。

**Width**

按鈕寬度。

**Image**

圖片顯示在按鈕上。

**Text**

設定文字顯示。

**Alignment**

設定對齊方式（左、中、右）。

**TextColor**

設定文字顏色。



## 事件

**Click()**

使用者點擊和放開按鈕時呼叫本事件。





Button 按鈕	<p><b>GotFocus()</b></p> <p>當指頭移到按鈕之上，代表現在可以點選本按鈕時呼叫本事件。</p> 
	<p><b>LostFocus()</b></p> <p>當指頭移出按鈕，代表不能點選本按鈕時呼叫本事件。</p> 
Canvas 畫布	圖形
	 <div style="float: right; border: 1px solid gray; padding: 2px;">component <b>Canvas1</b></div>
	功能
	<p>畫布為一矩形區域，可在其中執行繪畫等觸碰動作或設定動畫。</p> <p>在 Designer 或 Blocks Editor 中皆可設定畫布背景顏色、塗料顏色、背景圖片圖片、寬和高等屬性，注意寬和高的單位為像素必須為正值。</p> <p>畫布上的任何位置皆有一特定座標 (X,Y) 值，其中：</p> <ul style="list-style-type: none"> <li>• X 為座標點距離畫布左緣之距離，單位為像素。</li> <li>• Y 為座標點距離畫布上緣之距離，單位為像素。</li> </ul> <p>您可用畫布提供的事件來判斷畫布是否被觸摸或是動畫物件是否正在被拖動。另外也提供了畫點、線和圓的方法。</p>
屬性	
	<p><b>BackgroundColor</b></p> <p>設定畫布背景顏色。</p> 
	<p><b>BackgroundImage</b></p> <p>設定背景圖片。</p> 

Canvas  
畫布

Height

畫布高度。



LineWidth

畫線時的寬度。



PaintColor

畫線時的顏色。



Visible

本項需設為真，才能在螢幕上看到本元件。



Width

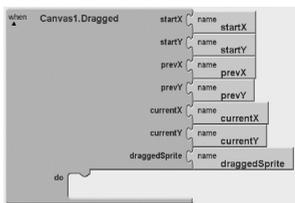
畫布寬度。



事件

Dragged(number startX, number startY, number prevX, number prevY, number currentX, number currentY, boolean draggedSprite)

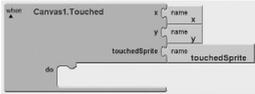
當使用者用手指頭拖拉時，觸控點會由 (prevX, prevY) 移到 (currentX, currentY)，當下的座標點皆是 (currentX, currentY)。(startX, startY) 這組座標代表使用者第一次觸碰螢幕時的那一點。"draggedSprite" 代表指定動畫元件正被使用者拖拉中。



Canvas  
畫布

**Touched(number x, number y, boolean touchedSprite)**

當使用者點擊畫布時，回傳 (x,y) 座標代表使用者所點擊的位置。如果「TouchedSprite」值為真代表某個動畫元件也正正好在此位置。



方法

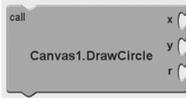
**Clear**

清除畫布上的各種塗鴉。如果畫布有背景圖片的話，本方法不會清除背景圖片。



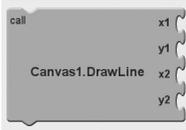
**DrawCircle(number x, number y, number r)**

在畫布上指定座標處 (x, y) 繪製一實心圓形，需指定其半徑 r。



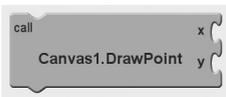
**DrawLine(number x1, number y1, number x2, number y2)**

在畫布上畫出一條直線，起始點 (x1, y1)，終點 (x2, y2)。



**DrawPoint(number x, number y)**

在畫布上指定座標處 (x, y) 畫出一個點。



**Save**

將畫布當下狀態存成一張圖檔，並儲存於 Android 裝置的外部儲存空間 (SD 記憶卡)，接著回傳該檔案的完整路徑。如果發生錯誤時，會由 Screen 元件的 ErrorOccurred 事件來處理。



Canvas  
畫布**SaveAs**

將畫布當下狀態存成一張圖檔，並儲存於 Android 裝置的外部儲存空間。本方法需指定存檔檔名，並必須加上副檔名為 .JPEG、.JPG 或 .PNG 其中之一。本方法一樣會回傳儲存檔案的完整路徑。

```
call Canvas1.SaveAs fileName
```

**DrawText**

在指定座標處 (x, y) 顯示文字 text 內容。

```
call Canvas1.DrawText text x y
```

**DrawTextAtAngle**

在指定座標處 (x, y) 顯示文字 text 內容，並指定旋轉角度 angle。angle 為數字型態，代表逆時針旋轉的角度，從 0 開始為水平。

```
call Canvas1.DrawTextAtAngle text x y angle
```

**GetPixelColor**

擷取指定座標處 (x, y) 的顏色，回傳值為數字，代表該處顏色的色碼。

```
call Canvas1.GetPixelColor x y
```

**GetBackGroundPixelColor**

擷取指定座標處 (x, y) 的顏色，回傳值為數字，代表該處顏色的色碼。本指令可擷取包含了 Canvas 畫布上大部份元件的顏色，包含點、線與圓圈，但不包含動畫元件。

```
call Canvas1.GetBackGroundPixelColor x y
```

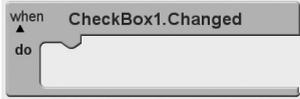
**SetBackGroundPixelColor**

設定指定座標處 (x, y) 的顏色，本指令與 DrawPoint 指令不同之處在於本指令可以指定顏色，DrawPoint 指令則無法指定顏色。

```
call Canvas1.SetBackGroundPixelColor x y color
```



Check box 檢查方塊	圖形	
	<input type="checkbox"/> Supersize	component <b>CheckBox1</b>
	功能	
	Checkbox 元件可以檢查使用者是否點選了它並以一個布林狀態來代表自己是否被點選。當使用者點選 Checkbox 元件會呼叫一事件來處理後續動作。我們可以在 Designer 或 Blocks Editor 中設定，有許多屬性可以改變 Checkbox 元件的外觀。	
	屬性	
	<b>BackgroundColor</b> 設定背景顏色。	
	set <b>CheckBox1.BackgroundColor</b> to <b>CheckBox1.BackgroundColor</b>	
	<b>Checked</b> 本項如果為真，代表使用者已點選本 Checkbox 元件。	
	set <b>CheckBox1.Checked</b> to <b>CheckBox1.Checked</b>	
	<b>Enabled</b> 如果設定為真，則可使用本元件。	
set <b>CheckBox1.Enabled</b> to <b>CheckBox1.Enabled</b>		
<b>Height</b> 元件高度 (y 軸像素)。		
set <b>CheckBox1.Height</b> to <b>CheckBox1.Height</b>		
<b>Width</b> 元件寬度 (x 軸像素)。		
set <b>CheckBox1.Width</b> to <b>CheckBox1.Width</b>		
<b>Text</b> 設定顯示文字。		
set <b>CheckBox1.Text</b> to <b>CheckBox1.Text</b>		

Check box 檢查方塊	<p><b>TextColor</b> 設定文字顏色。</p> 	
	<p><b>Visible</b> 本項需設為真，才能在螢幕上看到本元件。</p> 	
	<b>事件</b>	
	<p><b>Changed</b> CheckBox 點擊狀態改變時（使用者點擊或取消點擊）呼叫本事件。</p> 	
	<p><b>GotFocus</b> 當指頭移到 CheckBox 之上，代表現在可以點選本元件時呼叫本事件。</p> 	
	<p><b>LostFocus</b> 當指頭移出 CheckBox，代表不能點選本元件時呼叫本事件。</p> 	
Clock 時鐘	<b>圖形</b>	
		
	<b>功能</b>	
<p>Clock 元件可產生一個計時器，定期發起某個事件。它也可進行各種時間單位的運算與換算。Clock 元件的主要用途之一就是計時器( timer)。設定時間區件之後，計時器就會定期觸發，因而呼叫 timer 事件。</p> <p>Clock 元件的第二個用途是進行時間的各種運算，並以不同單位來表達時間。Clock 元件所使用的內部時間格式稱為 instant。Clock 元件的 Now 方法可以將現在的時間以 instant 來回傳。Clock 元件提供了許多方法來操作 instant，例如回傳一個短短數秒鐘或長達數月數年的 instant。此外它還提供了多種時間顯示方法，以指定 instant 的方式來顯示秒、分鐘、小時、天。</p>		



Clock  
時鐘

屬性

TimerInterval

時間區間，單位為毫秒。

set **Clock1.TimerInterval** to **Clock1.TimerInterval**

TimerEnabled

本項需設定為真，才可觸發計時器。

set **Clock1.TimerEnabled** to **Clock1.TimerEnabled**

TimerAlwaysFires

本項如果為真，即便 App Inventor 程式沒在螢幕前端，計時器依然會繼續觸發。

set **Clock1.TimerAlwaysFires** to **Clock1.TimerAlwaysFires**

事件

Timer()

計時器觸發時呼叫本事件。

when **Clock1.Timer**  
do

方法

SystemTime()

Android 裝置的內部系統時間，單位為毫秒。

Now()

從 Android 裝置的 clock 所讀取的當下現在時間。

MakeInstant(Text from)

以「MM/DD/YYYY hh:mm:ss」、「MM/DD/YYYY」或「hh:mm」等時間格式來產生一個 instant。

call **Clock1.MakeInstant** from

Clock  
時鐘**MakeInstantFromMillis(Number millis)**

以指定毫秒數來產生一個 instant。


 A block for the MakeInstantFromMillis method. It has a 'call' tab on the left and an 'instant' output on the right. The main text is 'Clock1.MakeInstantFromMillis' followed by a parameter 'millis'.
**GetMillis(instant)**

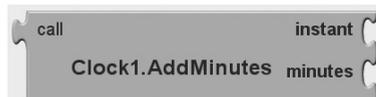
從 1970 年 1 月 1 日 0:00 開始累計的時間，單位為毫秒。


 A block for the GetMillis method. It has a 'call' tab on the left and an 'instant' input on the left. The main text is 'Clock1.GetMillis' followed by a parameter 'instant'.
**AddSeconds(instant, Number seconds)**

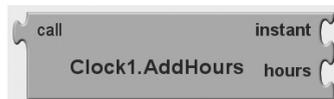
在指定 instant 後加上指定的秒數。


 A block for the AddSeconds method. It has a 'call' tab on the left and an 'instant' output on the right. The main text is 'Clock1.AddSeconds' followed by a parameter 'seconds'.
**AddMinutes(instant, Number minutes)**

在指定 instant 後加上指定的分鐘數。


 A block for the AddMinutes method. It has a 'call' tab on the left and an 'instant' output on the right. The main text is 'Clock1.AddMinutes' followed by a parameter 'minutes'.
**AddHours(instant, Number hours)**

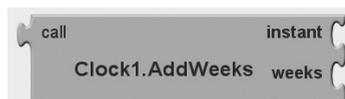
在指定 instant 後加上指定的小時數。


 A block for the AddHours method. It has a 'call' tab on the left and an 'instant' output on the right. The main text is 'Clock1.AddHours' followed by a parameter 'hours'.
**AddDays(instant, Number days)**

在指定 instant 後加上指定的天數。


 A block for the AddDays method. It has a 'call' tab on the left and an 'instant' output on the right. The main text is 'Clock1.AddDays' followed by a parameter 'days'.
**AddWeeks(instant, Number weeks)**

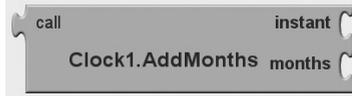
在指定 instant 後加上指定的周數。


 A block for the AddWeeks method. It has a 'call' tab on the left and an 'instant' output on the right. The main text is 'Clock1.AddWeeks' followed by a parameter 'weeks'.

Clock  
時鐘

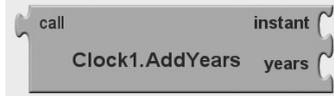
**AddMonths**(instant, Number months)

在指定 instant 後加上指定的月數。



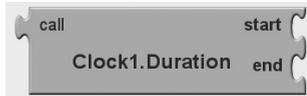
**AddYears**(instant, Number years)

在指定 instant 後加上指定的年數。



**Duration**(Calendar start, Calendar end)

兩個 instant 之間的時間差，單位為毫秒。



**Second**(Calendar instant)

一分鐘之內的秒數。



**Minute**(Calendar instant)

一小時之內的分鐘數。



**Hour**(Calendar instant)

一天中的小時數。



**DayOfMonth**(Calendar instant)

一個月中的某一天，由 1 ~ 31 之間的數字所代表。(整數)



Clock  
時鐘**Weekday(Calendar instant)**

一周中的某一天，由 1 ~ 7 之間的數來字代表星期一~星期日。(整數)

call **Clock1.Weekday** instant

**WeekdayName(Calendar instant)**

回傳今天是星期幾。(文字)

call **Clock1.WeekdayName** instant

**Month(Calendar instant) 月**

一個月中的某個月，由 1 ~ 12 之間的數來字代表 1 ~ 12 月。(整數)

call **Clock1.Month** instant

**MonthName(Calendar instant)**

回傳現在是幾月。(文字)

call **Clock1.MonthName** instant

**Year(Calendar instant)**

回傳現在是西元第幾年。(整數)

call **Clock1.Year** instant

**FormatDateTime(Calendar instant)**

回傳一字串代表某 instant 的時間日期格式。

call **Clock1.FormatDateTime** instant

**FormatDate(Calendar instant)**

回傳一字串代表某 instant 的日期格式。

call **Clock1.FormatDate** instant

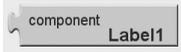
**FormatTime(Calendar instant)**

回傳一字串代表某 instant 的時間格式。

call **Clock1.FormatTime** instant



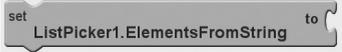
B

Image 圖片	圖形
	
	功能
	<p>Image 元件可用來顯示各種影像圖片，並可讓使用者點選或操作。Image 元件是用來顯示圖片的元件，您可以在 Designer 或 Blocks Editor 中指定該圖片的各種屬性。</p>
	屬性
	<p><b>Picture</b> 要在本 Image 元件中顯示的圖片。</p> 
<p><b>Visible</b> 本項需設為真，才能在螢幕上看到本元件。</p> 	
<p><b>Height</b> 元件高度（y 軸像素）。</p> 	
<p><b>Width</b> 元件寬度（x 軸像素）。</p> 	
Label 標籤	圖形
	
	功能
	<p>Label 元件可用來顯示文字。Label 元件可顯示在其 Text 屬性中所指定的文字。或者我們可以在 Designer 或 Blocks Editor 來調整文字的各種設定。</p>

屬性	
<b>BackgroundColor</b> 設定背景顏色。	
<b>FontBold</b> 設定文字粗體。	
<b>FontItalic</b> 設定文字斜體。	
<b>FontSize</b> 設定文字大小。	
<b>FontTypeface</b> 設定文字字形。	
<b>Height</b> 元件高度 (y 軸像素)。	
<b>Width</b> 元件寬度 (x 軸像素)。	
<b>Text</b> 設定顯示文字。	
<b>TextAlignment</b> 設定對齊方式 (左、中、右)。	
<b>TextColor</b> 設定文字的顏色。	

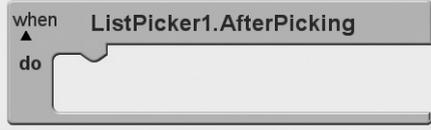
Label  
標籤

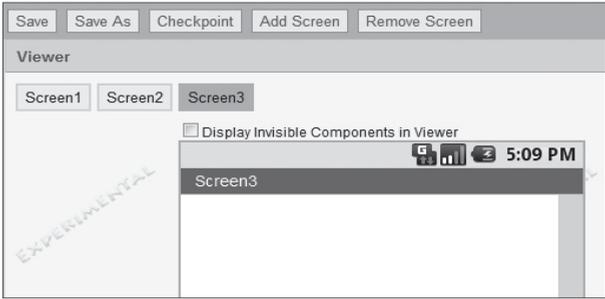


<p>Label 標籤</p>	<p><b>Visible</b> 本項需設為真，才能在螢幕上看到本元件。</p> 
<p>ListPicker 選取清單</p>	<div style="background-color: #cccccc; text-align: center; padding: 2px;">圖形</div>  <div style="background-color: #cccccc; text-align: center; padding: 2px;">功能</div> <p>使用者可點選 ListPicker 元件來選擇其中的某個項目，資料型態為字串陣列。</p> <p>當使用者點選 ListPicker 元件時，它會顯示一串項目讓使用者來選取。ListPicker 元件的項目可在 Designer 或 Block Editor 中設定 ElementsFromString 屬性 並以逗號分隔並排 (例如：choice 1, choice 2, choice)。或在 Blocks Editor 中將 ListPicker 元件的屬性指定為某個清單內容。</p> <p>其他屬性，包括文字對齊和背景顏色皆會影響 ListPicker 元件的外觀，我們也可設定其是否可以被點選 (Enabled)。</p> <div style="background-color: #cccccc; text-align: center; padding: 2px;">屬性</div> <p><b>Selection</b> 所選擇的清單項目。</p>  <p><b>Items</b> 將顯示 ListPicker 元件中的指定清單項目，項目彼此以逗號分隔。</p> <p><b>ElementsFromString</b> 將清單內容指定為 ListPicker 元件的項目。</p>  <p><b>Enabled</b> 本項需設定為真，才可使用本元件。</p> 

ListPicker 選取清單	<b>BackgroundColor</b> 設定背景顏色。 
	<b>FontBold</b> 設定文字粗體。
	<b>FontBold</b> 設定文字斜體。
	<b>FontSize</b> 設定文字大小。
	<b>FontTypeface</b> 設定文字字形。
	<b>Height</b> 元件高度 (y 軸像素)。 
	<b>Width</b> 元件寬度 (x 軸像素)。 
	<b>Text</b> 設定顯示文字。 
	<b>TextAlignment</b> 文字對齊方式 (左、中、右)。



ListPicker 選取清單	<p><b>TextColor</b> 設定 TextColor。</p> 
	<p><b>Visible</b> 本項需設為真，才能在螢幕上看到本元件。</p> 
	事件
	<p><b>AfterPicking()</b> 使用者點選 ListPicker 中某項目完成後呼叫本事件。</p> 
	<p><b>BeforePicking()</b> 使用者點選 ListPicker，但還沒點選某項目時呼叫本事件。</p> 
<p><b>GotFocus()</b> 當指頭移到 ListPicker 之上，代表現在可以點選本元件時呼叫本事件。</p> 	

	<p><b>LostFocus()</b> 當指頭移出 ListPicker，代表不能點選本元件時呼叫本事件。</p> 
Screen 螢幕	圖形
	
	功能
	<p>Screen 元件不會顯示在 palette 區中，您需要在 Designer 頁面中點選「Add Screen」來新增 Screen 元件。在早期的 App Inventor 版本中，每個程式都只能有一個 Screen 元件，且都叫做 Screen1。現在您可以根據自身的需要來新增 Screen 元件，不用都塞在一起了，是不是很方便呢？。</p> 
屬性	
<p><b>BackgroundColor</b> 設定背景顏色。</p> 	
<p><b>Image</b> 設定背景圖片。</p> 	



Screen  
螢幕

ScreenOrientation

Unspecified：螢幕會隨著握持方向而改變。  
Landscape：螢幕會鎖定在直向握持方向。  
Portrait：螢幕會鎖定在橫向握持方向。

.icon

當我們要把 App Inventor 程式下載到 Android 裝置上時，可由本屬性來設定本應用程式的圖示，建議使用 PNG 或 JPG 檔案，並請先調整圖檔解析度為 48 x48。  
注意：使用其它非 PNG 或 JPG 的圖檔例如 .ico 檔，可能會使 App Inventor 無法順利下載程式。

Scrollable

本屬性是在 Designer 中的一個 checkbox，如果點選，螢幕上將會有一個垂直捲軸，代表可上下捲動，應用程式的高度可以超過螢幕的實際 Y 軸解析度；反之未點選時，代表應用程式高度被限制設備的螢幕 Y 軸解析度。



Title

設定螢幕的標題，就是程式運行時螢幕左上角的那一串文字，一般來說都是 Screen1。比較常見的做法是將 Title 設為本應用程式的檔名，當然您也可以好好運用它，例如讓它顯示某些訊息或是運算結果。



Height

元件高度 (y 軸像素)。



Width

元件寬度 (x 軸像素)。



事件

**Initialize**

應用程式一啟動時就同步呼叫本事件，本事件可用來初始化某些變數以及執行一些前置性的操作。

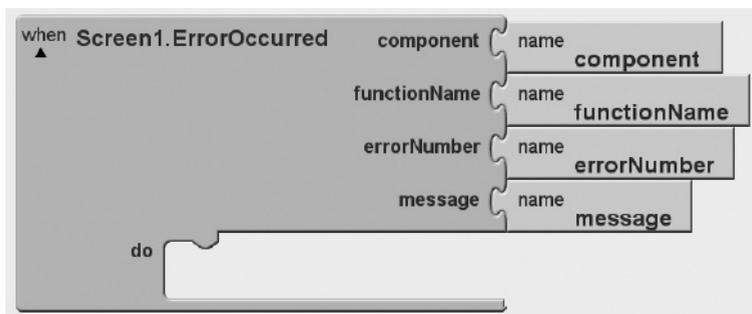


**ErrorOccurred(component component, text functionName, number errorNumber, test message)**

應用程式發生錯誤時呼叫本事件。ErrorOccurred 事件可處理的錯誤（不太多）包括：

- LEGO MINDSTORMS Nxt\* 等元件發生錯誤。
- Bluetooth 元件發生錯誤。
- Twitter 元件發生錯誤。
- SoundRecorder 元件發生錯誤。
- ActivityStarter 元件：當 StartActivity 被呼叫，但無正確對應之 Activity。
- LocationSensor 元件：當 LatitudeFromAddress 或 LongitudeFromAddress 失敗時。
- Player 元件：when 設置聲音屬性失敗。
- Sound 元件：當設置聲音元件屬性失敗或當播放功能失敗時。
- VideoPlayer 元件：當設置聲音屬性失敗時。

發生以上錯誤時，系統將以選單來顯示錯誤碼與錯誤訊息。您可以使用此事件處理器來預先處理各種異於上述的錯誤，這可藉由檢查錯誤碼（errorNumber）並採取適當的行動。





Password- TextBox	圖形
	
	功能
	<p>當使用者在 PasswordTextBox 元件中輸入密碼時，所有的輸入都會隱藏起來。</p> <p>PasswordTextBox 元件可說與 TextBox 元件完全一樣，只是會自動隱藏使用者的輸入內容。</p> <p>我們可以藉由 PasswordTextBox 元件的 Text 屬性來存取其內容。如果 Text 屬性為空白，您可以使用 Hint 屬性來建議使用者應該輸入的內容。Hint 屬性會以顏色較淡的文字顯示在 PasswordTextBox 元件中。</p> <p>PasswordTextBox 元件通常和按鈕元件搭配使用，使用者輸入密碼之後按下按鈕以執行後續動作。</p>
	屬性
	<p><b>BackgroundColor</b> 設定背景顏色。</p> 
	<p><b>Enabled</b> 本項需設定為真，才可使用本元件。</p> 
	<p><b>FontBold</b> 設定文字粗體。</p>
<p><b>FontItalic</b> 設定文字斜體。</p>	
<p><b>FontSize</b> 設定文字大小。</p>	
<p><b>FontTypeface</b> 設定文字字形。</p>	

Password-  
TextBox

**Height**  
高 (y-size)。



**Width**  
寬 (x-size)。



**TextAlignment**  
設定文字對齊方式 (左、中、右)。

**TextColor**  
設定文字顏色。



**Hint**  
密碼提示。



**Visible**  
本項需設為真，才能在螢幕上看到本元件。



**事件**

**GotFocus()**  
當指頭移到 PasswordTextBox 之上，代表現在可以點選本元件時呼叫本事件。



	<p><b>LostFocus()</b> 當指頭移出 PasswordTextBox，代表不能點選本元件時呼叫本事件。</p> 
	<p><b>圖形</b></p> 
TextBox	<p><b>功能</b></p> <p>使用者可在 TextBox 元件中輸入文字。</p> <p>TextBox 元件的初始值或是由使用者輸入的文字是由 Text 屬性所代表。如果 Text 屬性為空白，您可以使用 Hint 屬性來建議使用者應該輸入的內容。Hint 屬性會以顏色較淡的文字顯示在 TextBox 元件中。</p> <p>TextBox 元件的其它屬性可用來調整其外觀（例如 TextAlignment 與 BackgroundColor）以及是否可使用（Enabled）。</p> <p>TextBox 元件通常和按鈕元件搭配使用，使用者輸入內容之後按下按鈕以執行後續動作。如果您需要隱藏所輸入的內容，請使用 PasswordTextBox 元件。</p>
	<p><b>屬性</b></p>
	<p><b>BackgroundColor</b> 設定背景顏色。</p> 
	<p><b>Enabled</b> 本項需設定為真，才可使用本元件，意即是否可輸入文字。</p> 
	<p><b>FontBold</b> 設定文字粗體。</p>
	<p><b>FontItalic</b> 設定文字斜體。</p>

TextBox	<b>FontSize</b> 設定文字大小。
	<b>FontTypeface</b> 設定文字字形。
	<b>NumbersOnly</b> 本項如果設定為真，代表只能輸入數字。 
	<b>Multiline</b> 本項如果設定為真，代表可輸入多行內容，使用 Enter 或 Return 鍵換行。 
	<b>Text</b> 設定顯示文字。 
	<b>Height</b> 元件高度 (y 軸像素)。 
	<b>Width</b> 元件寬度 (x 軸像素)。 
	<b>TextAlignment</b> 設定為字對齊方式 (左、中、右)。
	<b>TextColor</b> 設定文字顏色。 



TextBox	<p><b>HideKeyboard</b> 隱藏螢幕鍵盤。</p> <p>call <b>TextBox1.HideKeyboard</b></p>
	<p><b>Hint</b> 內容提示。</p> <p>set <b>TextBox1.Hint</b> to <b>TextBox1.Hint</b></p>
	<p><b>Visible</b> 本項需設為真，才能在螢幕上看到本元件。</p> <p>set <b>TextBox1.Visible</b> to <b>TextBox1.Visible</b></p>
	事件
	<p><b>GotFocus()</b> 當指頭移到 TextBox 之上，代表現在可以點選本元件時呼叫本事件。</p> <p>when <b>TextBox1.GotFocus</b></p> <p>▲</p> <p>do</p>
<p><b>LostFocus()</b> 當指頭移出 TextBox，代表不能點選本元件時呼叫本事件。</p> <p>when <b>TextBox1.LostFocus</b></p> <p>▲</p> <p>do</p>	

## 圖形



## 功能

TinyDB 元件可用來儲存資料，之後每次運行應用程式時都可使用 TinyDB 元件的資料。

TinyDB 元件為一非可視元件。由 App Inventor 所編寫的應用程式每次執行時都會重新被初始化。如果程式對某個變數進行調整之後退出程式，則下一次執行程式時該變數的值將恢復原狀。TinyDB 對於應用程式來說是一個永久的資料儲存器，意即每次程式啟動時都可以使用它所包含的資料。例如您可以保存遊戲的最高分排行榜，每次玩遊戲時都可顯示這一筆資料。

不同的資料項目是根據標籤 (tag) 來儲存。每當儲存一筆資料時，您需要指定這筆資料的標籤。因此您可依據這個標籤來取用這筆資料。如果某個標籤下沒有任何資料，則回傳值為一個空的字串。反之，您可藉由回傳值是否為空字串來判斷某個標籤下是否有資料，例如沒有輸入任何東西的 TextBox。

每個應用程式只能有一個資料存儲區。如果您有多個 TinyDB 元件，它們將使用相同的資料存儲區。如果要使用多個資料存儲區，您需要使用不同的金鑰 (key)。再者，每個應用程式都有它專屬的資料存儲區，因此無法使用 TinyDB 元件讓 Android 裝置上的兩個不同應用程式彼此傳遞資料。

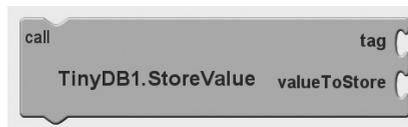
屬性：無

事件：無

## 方法

**StoreValue(Text tag, valueToStore)**

在指定標籤下的儲存一筆資料，tag 參數必須是為文字字串；valueToStore 可以為字串或清單。

**GetValue(Text tag)**

取得指定標籤下的資料。如果其下沒有任何資料，則傳回空的字串。





## B-2 Media Component 多媒體元件

- Camera
- ImagePicker
- Player
- Sound
- VideoPlayer

Camera 照相機	圖形
	 <div style="border: 1px solid gray; padding: 5px; display: inline-block;">component <b>Camera1</b></div>
	功能
	<p>Camera 元件可呼叫 Android 裝置上的相機進行拍照。</p> <p>Camera 元件為一非可視元件，它可呼叫 Android 裝置上的相機進行拍照。拍完照之後，您可從 AfterPicture 事件中的參數找到剛剛所拍照片的檔案位置。您可將這個檔案位置用於 Image 元件的 Picture 屬性，將這個 Image 的圖片指定為剛剛所拍的照片。</p>
	屬性：無
	<p><b>TakePicture()</b></p> <p>啟動 Android 裝置上的相機並進行拍照。</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;">       call <b>Camera1.TakePicture</b> </div>
	事件
	<p><b>AfterPicture(text image)</b></p> <p>拍照完成後呼叫本事件，image 這個字串參數代表剛剛所拍照片儲存於 Android 裝置中的位置，可用來呼叫這張照片。</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;">       when <b>Camera1.AfterPicture</b> image name <b>image</b>        do     </div>

Imagepicker  
圖片選取器

## 圖形

component

ImagePicker1

## 功能

ImagePicker 元件可從您的圖片庫中選取圖片。

ImagePicker 是一種特殊的 ListPicker，專門用來選取圖片，其內容會自動指定為模擬器或 Android 裝置上的圖片庫。當您點選它之後，會跳到 Android 裝置上的圖片庫，請接著選擇您所需要的圖片。當您選擇好圖片之後，ImagePath 這個屬性是用一個字串來代表該圖片的路徑。您可使用該參數來設定按鈕的背景圖片。

## 屬性

## ImagePath

使用者所選擇的圖片，以字串回傳該圖片的位置。

A block with a tab labeled 'ImagePicker1' and the text 'ImagePath'.

## BackgroundColor

設定背景顏色。

A block with a 'set' tab, the text 'ImagePicker1.BackgroundColor', a 'to' tab, and another 'ImagePicker1.BackgroundColor'.

## Enabled

本項需設定為真，才可使用本元件。

A block with a 'set' tab, the text 'ImagePicker1.Enabled', a 'to' tab, and another 'ImagePicker1.Enabled'.

## FontBold

設定文字粗體。

## FontItalic

設定文字斜體。

## FontSize

設定文字大小。



Imagepicker  
圖片選取器

<b>FontTypeface</b> 設定文字字形。
<b>Image</b> 設定背景圖片。 
<b>Text</b> 設定顯示文字。 
<b>TextAlignment</b> 設定文字對齊方式（左、中、右）。
<b>TextColor</b> 設定文字顏色。 
<b>Visible</b> 本項需設為真，才能在螢幕上看到本元件。 
<b>Height</b> 元件高度（y 軸像素）。 
<b>Width</b> 元件寬度（x 軸像素）。 

Imagepicker  
圖片選取器

## 事件

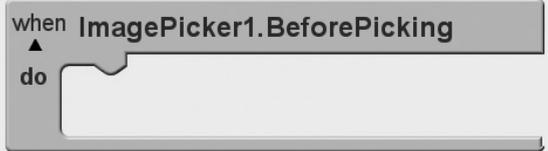
## AfterPicking

使用者點選 ImagePicker 中某項目完成後呼叫本事件。



## BeforePicking

使用者點選 ImagePicker，但還沒點選某項目時呼叫本事件。



## GotFocus

當指頭移到 ImagePicker 之上，代表現在可以點選本元件時呼叫本事件。



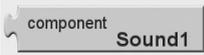
## LostFocus

當指頭移出 ImagePicker，代表不能點選本元件時呼叫本事件。





Imagepicker 圖片選取器	方法
	<p><b>Open</b> 開啟 Imagepicker，效果同使用者親自點選一樣。</p> 
Player 播放器	圖形
	 
	功能
	<p>Player 元件可播放聲音或影像檔，或使裝置震動。</p> <p>Player 元件為一非可視元件，可播放聲音或影像檔和控制震動手機。要播放的檔案名稱是從 Source 屬性中設定，這可以在 Designer 或 Block Editor 中設定。震動的時間長度則是在 Block Editor 中設定，單位為毫秒。</p> <p>請參考 <a href="http://developer.android.com/guide/appendix/media-formats.html">http://developer.android.com/guide/appendix/media-formats.html</a> 來參考有關檔案類型的詳細資訊。</p> <p>Player 元件主要用在播放較長的聲音 / 影像檔，或使裝置震動，如果播放較短的聲音檔時請使用 Sound 元件。</p>
	屬性
<p><b>Source</b> 指定要播放的聲音檔。</p> 	
	方法
	<p><b>Pause</b> 暫停播放聲音或影片。</p> 
	<p><b>Start</b> 開始播放聲音或影片。</p> 

Player 播放器	<p><b>Stop</b> 停止播放聲音或影片。</p> 
	<p><b>Vibrate(number milliseconds)</b> 使手機震動，需指定震動時間，單位為毫秒。</p> 
Sound 聲音	<b>圖形</b>
	 
	<b>功能</b>
	<p>Sound 元件可用來播放較短的音效檔，或使裝置震動。</p> <p>Sound 為一非可視元件，它可用來播放音效檔和讓手機震動（單位為毫秒）。要播放的音效檔檔名可在 Designer 或 Block Editor 中設定。</p> <p>請參考 <a href="http://developer.android.com/guide/appendix/media-formats.html">http://developer.android.com/guide/appendix/media-formats.html</a> 來參考有關檔案類型的詳細資訊。</p> <p>Sound 元件適用於播放較短的聲音檔，如果要播放較長的聲音檔例如一首歌，這時請使用 Player 元件。</p>
	<b>屬性</b>
<p><b>Source</b> 指定要播放的音效檔。</p> 	
<p><b>MinimumInterval</b> 播放時間長度，單位為毫秒。</p> 	



Sound  
聲音

方法

Pause

暫停播放音效檔

call **Sound1.Pause**

Play

開始播放音效檔。

call **Sound1.Play**

Resume

回復先前狀態。

call **Sound1.Resume**

Stop

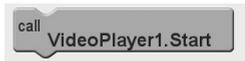
停止播放音效檔。

call **Sound1.Stop**

Vibrate(number millisecs)

使手機震動，需指定震動時間，單位為毫秒。

call **Sound1.Vibrate** millisecs

圖形	
	
功能	
<p>VideoPlayer 元件可用來播放影像檔。</p> <p>VideoPlayer 元件是一個可播放影片的多媒體元件，它會在裝置螢幕上顯示為一個矩形。使用者點擊矩形時會出現影像控制工具列：播放 / 暫停（play/pause）、跳回上一個檔案（skip ahead）與跳到下一個檔案（skip backward）。我們可以在應用程式中藉由開始（Start）、暫停（Pause）和跳到影片檔中的指定時間（SeekTo）等方法來控制重播相關動作。</p> <p>可播放的影像檔格式可為 wmv、3gp 或 mp4 等格式。有關格式的更多詳細資訊，請參考 <a href="http://developer.android.com/guide/appendix/media-formats.html">http://developer.android.com/guide/appendix/media-formats.html</a>。</p> <p>App Inventor 接受的影像檔最大為 1MB，單一應用程式中所有的影像檔大小總和不得超過 5MB。如果您的檔案太大，安裝時可能會發生錯誤。這時請減少影像檔數量或它們的大小。您可以利用 Windows Movie Maker 或蘋果的 iMovie 等影像編輯軟體來剪接或轉成其他檔案格式。</p>	
屬性	
<p><b>Source</b></p> <p>指定要播放的影像檔。</p> 	
<p><b>Visible</b></p> <p>本項需設為真，才能在螢幕上看到本元件。</p> 	
方法	
<p><b>Pause</b></p> <p>暫停播放影像檔。</p> 	
<p><b>Start</b></p> <p>開始播放影像檔。</p> 	

VideoPlayer  
影像播放器



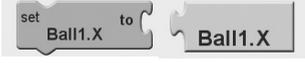
VideoPlayer 影像播放器	<p><b>SeekTo(number millisecs)</b> 跳到影片檔中的指定時間，單位為毫秒。</p> <p>call <b>VideoPlayer1.SeekTo</b> ms</p>
	<p><b>GetDuration</b> 回傳影片檔之時間長度，單位為毫秒。</p> <p>call <b>VideoPlayer1.GetDuration</b></p>
	<p>事件</p> <p><b>Completed</b> 當影片檔播放完畢後呼叫本事件。</p> <p>when <b>VideoPlayer1.Completed</b> do</p>

## B-3 Animation 動畫元件

Ball  
ImageSprite

Ball 球	圖形
	 <div style="float: right; border: 1px solid gray; padding: 2px;">component <b>Ball1</b></div>
	功能

Ball 元件為一球型的特殊動畫元件。  
 Ball 元件為一可置放於畫布中的圓形動畫元件，當它被觸碰、拖拉、與其他動畫元件 (ImageSprite 或 Ball) 互動時或與畫布邊緣接觸時，它可根據不同事件執行對應動作。  
 Ball 元件也可依照其屬性自行移動，例如要讓一個 Ball 元件每 500 毫秒往畫布上緣移動 4 個像素，您可以設置 Speed 屬性設為 4，Interval 屬性為 500，Heading 屬性為 90 (度) 和 Enabled 屬性為真，您可以調整這些屬性來改變球的行為。  
 Ball 元件和 Imagesprite 之間的不同就是 Imagesprite 可以透過上傳圖檔來改變其外觀，但 Ball 元件只能調顏色 (PaintColor) 和大小 (Radius)。

屬性	
Ball 球	<p><b>Radius</b> 半徑 球的半徑。</p>  <p>set Ball1.Radius to Ball1.Radius</p>
	<p><b>PaintColor</b> 畫筆顏色 球的顏色。</p>  <p>set Ball1.PaintColor to Ball1.PaintColor</p>
	<p><b>Enabled</b> 啟用 本項需設定為真，球才可移動。</p>  <p>set Ball1.Enabled to Ball1.Enabled</p>
	<p><b>Interval</b> 時間區間 球的移動頻率，單位為毫秒。</p>  <p>set Ball1.Interval to Ball1.Interval</p>
	<p><b>Visible</b> 可見的 本項需設定為真，球才能顯示在螢幕上。</p>  <p>set Ball1.Visible to Ball1.Visible</p>
	<p><b>Heading</b> 方向 球的移動方向。</p>  <p>set Ball1.Heading to Ball1.Heading</p>
	<p><b>X</b> 球的 X 座標。</p>  <p>set Ball1.X to Ball1.X</p>



Ball  
球

Y

球的 Y 座標。

Speed 速度

球每單位時間之移動距離，單位為像素。

事件

CollidedWith(component other)

當兩個動畫元件（動畫或球）相撞時呼叫本事件，參數 other 代表在本次碰撞中的另一個元件。

Dragged(number startX, number startY, number prevX, number prevY, number currentX, number currentY)

球被拖移時呼叫本事件，參數說明請參閱 Canvas 元件。

when Ball1.Dragged

- startX name startX
- startY name startY
- prevX name prevX
- prevY name prevY
- currentX name currentX
- currentY name currentY

do

Ball  
球

EdgeReached(number edge)



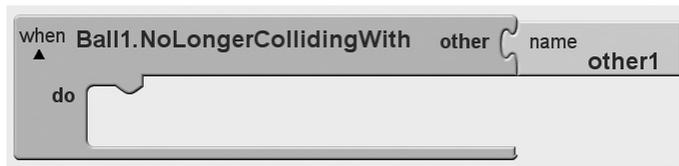
當球與螢幕邊緣接觸時呼叫本事件，參數 `edge` 代表球接觸的位置，如下所示：

- north = 1，螢幕上（北）緣
- northeast = 2，螢幕右上（東北）角
- east = 3，螢幕右（東）緣
- southeast = 4，螢幕右下（東南）角
- south = -1，螢幕下（南）緣
- southwest = -2，螢幕左下（西南）角
- west = -3，螢幕左（西）緣
- northwest = -4，螢幕左上（西北）角

請注意相反的方向是彼此互為相反數，如右圖

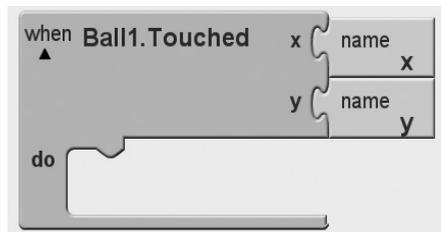
NoLongerCollidingWith(component other)

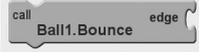
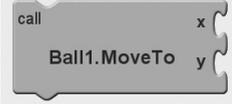
當兩個動畫元件不再碰撞時呼叫本事件。



Touched(number x, number y)

當球被點擊時呼叫本事件。



Ball 球	方法	
	<p><b>Bounce(number edge)</b> 使球彈跳，就好像真的撞到牆或角落一樣。參數和 EdgeReached 事件的參數相同，因此我們可以利用 EdgeReached 事件讓球每次碰到畫布邊緣都會彈跳，讓球栩栩如生地自由彈跳。</p>	
		
	<p><b>boolean CollidingWith(component other)</b> 代表球是否和指定元件發生碰撞。</p>	
		
	<p><b>MoveIntoBounds</b> 如果球跑出界了，可利用本方法將它抓回界內。</p>	
		
<p><b>MoveTo(number x,number y)</b> 讓球移動到指定點座標。</p>		
		
<p><b>PointTowards(component target)</b> 讓球轉向對準指定的目標。新的 heading 即為兩個元件中心所構成直線之指向。</p>		
		
ImageSprite 動畫	圖形	
		
	功能	
<p>ImageSprite 元件為一個動畫物件，它可和畫布上，的球和其他 ImageSprite 進行互動。 ImageSprite 元件為一可置放於畫布中的動畫物件，可以回應觸碰和拖拉事件，與其他動畫物件（球或其他 ImageSprite）和畫布邊緣互動，或者它可根據屬性設定來移動。例如要讓一個 ImageSprite 元件每秒鐘往左邊移動 10 個像素，您可以將 Speed 屬性設為 4，Interval 屬性為 1000（毫秒），Heading 屬性為 180（度）和 Enabled 屬性為真，您可以自由調整這些屬性來改變 ImageSprite 的行為。</p>		

ImageSprite  
動畫

## 屬性

**Picture 圖片**

設定 Imagesprite 元件的圖片。

**Enabled 啟用**

本項需設為真，Imagesprite 元件才可移動。

**Interval 時間區間**

Imagesprite 元件的動作頻率，單位為毫秒。

**Rotates 旋轉**

本項如果是真，Imagesprite 元件的圖片的 image 會旋轉直到對齊 sprite 的 heading 屬性為止。反之，即便 sprite 的 heading 改變，image 也不會跟著旋轉。

**Visible 可見**

本項需設為真，才能在螢幕上看到 ImageSprite 元件。

**Heading 指向**

ImageSprite 的旋轉方向，單位為度。0 度代表水平指向右（東）方；90 度是朝上（北）方，180 度是左（西）方，270 度是下（南）方。

**X**

ImageSprite 的 X 座標，0 為畫布左緣。



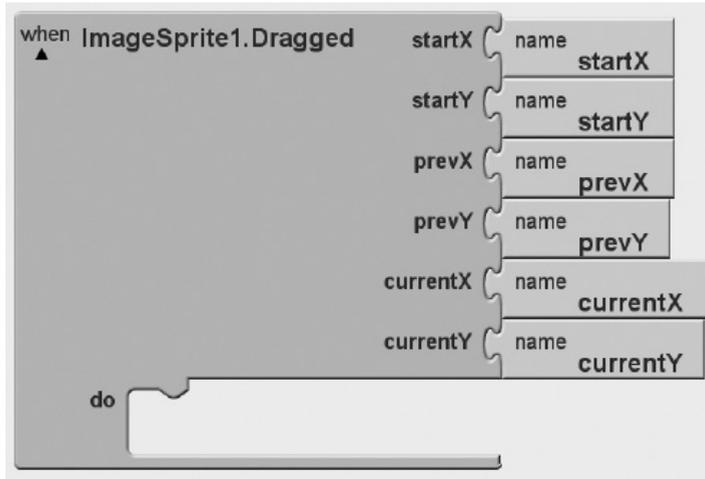


ImageSprite 動畫	<p><b>Y</b></p> <p>ImageSprite 的 y 座標，0 為畫布上緣。</p> 
	<p><b>Speed 速度</b></p> <p>ImageSprite 每單位時間移動的像素。</p> 
	<p><b>Width 寬</b></p> <p>ImageSprite 的寬。</p> 
	<p><b>Height 高</b></p> <p>ImageSprite 的高。</p> 
	事件
	<p><b>CollidedWith(component other)</b></p> <p>當兩個動畫元件(動畫或球)相撞時呼叫本事件。參數 other 代表與它碰撞的另一個元件。</p> 

ImageSprite  
動畫

**Dragged(number startX, number startY, number prevX, number prevY, number currentX, number currentY)**

Sprite 被拖移時呼叫本事件，參數說明請參閱 Canvas 元件。



**EdgeReached(number edge)**

當 ImageSprite 與螢幕邊緣接觸時呼叫本事件，參數 **edge** 代表球接觸的位置，如下所示：

- north = 1，螢幕上（北）緣
- northeast = 2，螢幕右上（東北）角
- east = 3，螢幕右（東）緣
- southeast = 4，螢幕右下（東南）角
- south = -1，螢幕下（南）緣
- southwest = -2，螢幕左下（西南）角
- west = -3，螢幕左（西）緣
- northwest = -4，螢幕左上（西北）角

請注意相反的方向是彼此互為相反數。





ImageSprite  
動畫

NoLongerCollidingWith(component other)

當兩個動畫元件不再碰撞時呼叫本事件。

Touched(number x, number y)

當 ImageSprite 球被點擊時呼叫本事件。

方法

Bounce(number edge)

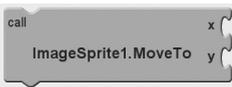
使 ImageSprite 彈跳，就好像真的撞到牆或角落一樣。參數和 EdgeReached 事件的參數相同，因此我們可以利用 EdgeReached 事件讓球每次碰到畫布邊緣都會彈跳，讓 ImageSprite 栩栩如生地自由彈跳。

boolean CollidingWith(component other)

代表 ImageSprite 是否和指定元件發生碰撞。

MoveIntoBounds()

如果 ImageSprite 跑出界了，可利用本方法將它抓回界內。

ImageSprite 動畫	<p><b>MoveTo(number x,number y)</b> 讓 ImageSprite 移動到指定點座標。</p> 
	<p><b>PointTowards(component target)</b> 讓 ImageSprite 轉向對準指定的目標。新的 heading 即為兩個元件中心所構成直線之指向。</p> 

## B-4 Social Components 通訊元件

- ContactPicker
- EmailPicker
- PhoneCall
- PhoneNumberPicker
- Texting
- Twitter

ContactPicker 聯絡人 選擇器	圖形	
		
	功能	
<p>ContactPicker 元件可讓使用者選擇 Android 通訊錄的一筆資料。 ContactPicker 元件是一個用來顯示裝置上通訊錄的按鈕，並可讓使用者選擇特定內容。當使用者點選之後，請參考下列屬性：</p> <ul style="list-style-type: none"> <li>• ContactName：聯絡人姓名。</li> <li>• EmailAddress：聯絡人的主要電子郵件。</li> <li>• Picture：聯絡人大頭照，照片 property value for the Image or Image Sprite component.</li> </ul> <p>尚有其他屬性可用來設定ContactPicker元件之外觀，包括文字對齊、背景圖片以及事件（是否可點選）。</p>		



屬性	
Enabled	設定是否可以使用。
set	
Image	設定圖片。
set	
BackgroundColor	設定背景顏色。
set	
ContactPicker 聯絡人 選擇器	
ContactName	聯絡人名稱。
EmailAddress	聯絡人的電子郵件地址。
Picture	聯絡人的圖片。
FontBold	設定文字粗體。

ContactPicker  
聯絡人  
選擇器**FontItalic**

設定文字斜體。

**FontSize**

設定文字大小。

**FontTypeface**

設定文字字形。

**Text**

設定文字顯示。

set **ContactPicker1.Text** to **ContactPicker1.Text**

**TextAlignment**

設定對其方式（左、中、右）。

**TextColor**

設定文字的顏色。

set **ContactPicker1.TextColor** to **ContactPicker1.TextColor**

**Visible**

本項需設為真，才能在螢幕上看到本元件。

set **ContactPicker1.Visible** to **ContactPicker1.Visible**

**Width**

元件寬度（x 軸像素）。

set **ContactPicker1.Width** to **ContactPicker1.Width**

**Height**

元件高度（y 軸像素）。

set **ContactPicker1.Height** to **ContactPicker1.Height**



ContactPicker  
聯絡人  
選擇器

事件

AfterPicking

使用者點選 ContactPicker 中某項目完成後呼叫本事件。代表使用者選取了某位聯絡人。

when **ContactPicker1.AfterPicking**

do

BeforePicking

使用者點選 ContactPicker，但還沒點選某項目時呼叫本事件。

when **ContactPicker1.BeforePicking**

do

GotFocus

當指頭移到 ContactPicker 之上，代表現在可以點選本元件時呼叫本事件。

when **ContactPicker1.GotFocus**

do

LostFocus

當指頭移出 ContactPicker，代表不能點選本元件時呼叫本事件。

when **ContactPicker1.LostFocus**

do

方法

Open

開啟 ContactPicker，效果同使用者親自點選一樣。

call **ContactPicker1.Open**

	圖形
	
	功能
	<p>EmailPicker 元件可讓使用者從 Android 裝置上的通訊錄輸入一個電子郵件信箱。</p> <p>EmailPicker 元件就是一個 textbox，使用者可輸入研究信一個電子郵件信箱，且支援自動完成功能（auto-completion）。EmailPicker 元件的初始值與輸入後的值都可以在 Text 屬性中設定。如果 Text 屬性一開始為空，則會 Hint 屬性會以顏色較淡的文字顯示在 EmailPicker 元件中。</p> <p>其他屬性，包括文字對齊和背景顏色皆會影響 ListPicker 元件的外觀，我們也可設定其是否可以被點選（Enabled）。</p> <p>EmailPicker 元件通常是當作按鈕來使用，使用者輸入完畢之後須點選它才能執行後續動作。</p>
	屬性
EmailPicker 電子郵件 選擇器	<p><b>Enabled</b> 本項需設定為真，才可使用本元件。</p> 
	<p><b>BackgroundColor</b> 設定背景顏色。</p> 
	<p><b>FontBold</b> 設定文字粗體。</p>
	<p><b>FontItalic</b> 設定文字斜體。</p>
	<p><b>FontSize</b> 設定文字大小。</p>
	<p><b>FontTypeface</b> 設定文字字形。</p>



EmailPicker  
電子郵件  
選擇器

**Text**

設定顯示文字。

set EmailPicker1.Text to EmailPicker1.Text

**TextAlignment**

設定文字對齊（左、中、右）。

**TextColor**

設定文字顏色。

set EmailPicker1.TextColor to EmailPicker1.TextColor

**Hint**

內容提示。

set EmailPicker1.Hint to EmailPicker1.Hint

**Visible**

本項需設為真，才能在螢幕上看到本元件。

set EmailPicker1.Visible to EmailPicker1.Visible

**Height**

元件高度（y 軸像素）。

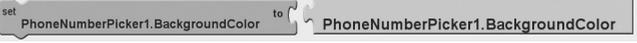
set EmailPicker1.Height to EmailPicker1.Height

**Width**

元件寬度（x 軸像素）。

set EmailPicker1.Width to EmailPicker1.Width

EmailPicker 電子郵件 選擇器	事件
	<p><b>GotFocus</b></p> <p>當指頭移到 EmailPicker 之上，代表現在可以點選本元件時呼叫本事件。</p> 
	<p><b>LostFocus</b></p> <p>當指頭移出 EmailPicker，代表不能點選本元件時呼叫本事件。</p> 
PhoneCall 打電話	圖形
	 
	功能
	<p>PhoneCall 元件可用來撥打電話。</p> <p>PhoneCall 元件為一非可視元件，它可向 PhoneNumber 屬性中所設定的電話號碼實際地打一通電話，這可以在 Designer 或 Block Editor 中設定。您可以使用 MakePhoneCall 方法來從應用程式中撥打一通電話。</p> <p>PhoneCall 元件通常會和 ContactPicker 元件搭配使用，使用者從裝置上的通訊錄選擇完之後即可將選擇內容指定為 PhoneCall 元件的 PhoneNumber 屬性內容。</p> <p>如果您需要直接指定電話號碼，請在 PhoneNumber 屬性中輸入正確的號碼格式（例如 022xxxxxxx 或 09xxxxxxx）。電話號碼中可包含「-」、「_」或句號，這些會自動被忽略，但請注意不可包含空格。</p> <p>09XX-XXX-XXX →合法格式 09XX XXX XXX →無法辨識</p>
	屬性
	<p><b>PhoneNumber</b></p> <p>要撥打的電話號碼。</p> 

<p>PhoneCall 打電話</p>	<p style="text-align: center;">方法</p> <p><b>MakePhoneCall</b> 對 PhoneNumber 屬性中指定的電話號碼撥打一通電話。</p> 
<p>PhoneNumber- Picker 電話號碼 選取器</p>	<p style="text-align: center;">圖形</p>  <p style="text-align: center;">功能</p> <p>PhoneNumberPicker 元件可讓使用者從 Android 裝置通訊錄中選擇某個聯絡人的電話號碼。當使用者點選 PhoneNumberPicker 元件時，螢幕會跳到 Android 裝置通訊錄。點選完畢之後，下列屬性就會根據所選擇的聯絡人自動完成：</p> <ul style="list-style-type: none"> <li>• ContactName：聯絡人姓名。</li> <li>• PhoneNumber：所選聯絡人的電話號碼。</li> <li>• EmailAddress：所選聯絡人的主要電子郵件信箱。</li> <li>• Picture：聯絡人大頭照，這可以當作 Image 或 ImageSprte 元件的 Picture 屬性。</li> </ul> <p>其他屬性，包括文字對齊和背景顏色皆會影響 PhoneNumberPicker 元件的外觀，我們也可設定其是否可以被點選 (Enabled)。</p> <p style="text-align: center;">屬性</p> <p><b>Enabled</b> 本項需設定為真，才可使用本元件。</p>  <p><b>Image</b> 設定顯示圖片。</p> 
	<p><b>BrackgroundColor</b> 設定背景顏色。</p> 

PhoneNumber-  
Picker  
電話號碼  
選取器

**ContactName**

選擇聯絡人姓名。

**PhoneNumberPicker1.ContactName**

**EmailAddress**

選擇聯絡人電子郵件地址。

**PhoneNumberPicker1.EmailAddress**

**PhoneNumber**

選擇聯絡人的手機號碼。

**PhoneNumberPicker1.PhoneNumber**

**Picture**

選擇聯絡人的照片。

**PhoneNumberPicker1.Picture**

**FontBold**

設定文字粗體。

**FontItalic**

設定文字斜體。

**FontSize**

設定文字大小。

**FontTypeface**

設定文字字形。

**Text**

設定顯示文字。

set **PhoneNumberPicker1.Text** to **PhoneNumberPicker1.Text**



PhoneNumberPicker  
電話號碼  
選取器

**TextAlignment**

設定文字對齊方式（左、中、右）。

**TextColor**

設定文字顏色。

```

set PhoneNumberPicker1.TextColor to PhoneNumberPicker1.TextColor
  
```

**Visible**

本項需設為真，才能在螢幕上看到本元件。

```

set PhoneNumberPicker1.Visible to PhoneNumberPicker1.Visible
  
```

**Height**

元件高度（y 軸像素）。

```

set PhoneNumberPicker1.Height to PhoneNumberPicker1.Height
  
```

**Width**

元件寬度（x 軸像素）。

```

set PhoneNumberPicker1.Width to PhoneNumberPicker1.Width
  
```

方法

**Open**

開啟 PhoneNumberPicker，效果同使用者親自點選一樣。

```

call PhoneNumberPicker1.Open
  
```

事件

**AfterPicking**

使用者點選 PhoneNumberPicker 中某項目完成後呼叫本事件。

```

when PhoneNumberPicker1.AfterPicking
do
  
```

PhoneNumber-  
Picker  
電話號碼  
選取器

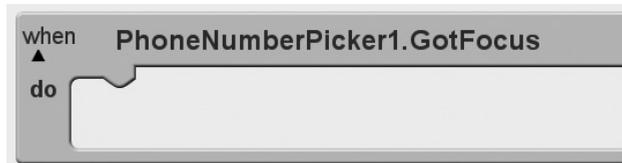
#### BeforePicking

使用者點選 PhoneNumberPicker，但還沒點選某項目時呼叫本事件。



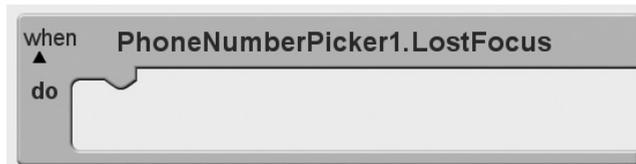
#### GotFocus

當指頭移到 PhoneNumberPicker 之上，代表現在可以點選本元件時呼叫本事件。



#### LostFocus

當指頭移出 PhoneNumberPicker，代表不能點選本元件時呼叫本事件。



### 圖形



### 功能

Texting 元件可讓使用者收發簡訊。

當呼叫 `SendMessage` 方法時，Texting 元件會對 `PhoneNumber` 屬性中所指定的電話號碼送出一封簡訊（簡訊內容是在 `Message` 屬性中設定）。除非我們將 `ReceivingEnabled` 屬性設為 `false`，不然 Texting 元件可以持續接收文字簡訊。當收到簡訊時，會自動呼叫 `MessageReceived` 事件並回傳寄件人號碼與訊息內容。

Texting 元件通常會和 `ContactPicker` 元件搭配使用，使用者從裝置上的通訊錄選擇完之後即可將選擇內容指定為 Texting 元件的 `PhoneNumber` 屬性內容。

如果您需要直接指定電話號碼，請在 `PhoneNumber` 屬性中輸入正確的號碼格式（例如 `022XXXXXXX` 或 `09XXXXXXX`）。電話號碼中可包含「-」、「\_」或句號，這些會自動被忽略，但請注意不可包含空格。

Texting  
簡訊

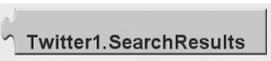


Texting 簡訊	屬性
	<p><b>PhoneNumber</b> 欲發送簡訊的電話號碼。</p> <p>set <b>Texting1.PhoneNumber</b> to <b>Texting1.PhoneNumber</b></p>
	<p><b>Message</b> 欲發送的簡訊內容。</p> <p>set <b>Texting1.Message</b> to <b>Texting1.Message</b></p>
	<p><b>ReceivingEnabled</b> 本項需設為 true，Texting 元件才可以接收簡訊。</p> <p>set <b>Texting1.ReceivingEnabled</b> to <b>Texting1.ReceivingEnabled</b></p>
	事件
	<p><b>MessageReceived(Text number, Text message Text)</b> 收到簡訊時呼叫本事件，參數 number 代表寄件人電話號碼，messageText 代表簡訊內容。</p> <p>when <b>Texting1.MessageReceived</b> number name <b>number</b> messageText name <b>messageText</b></p> <p>do</p>
	方法
	<p><b>SendMessage</b> 向 PhoneNumber 屬性中所指定的電話號碼發送一封簡訊，簡訊內容是在 Message 屬性中設定。</p> <p>call <b>Texting1.SendMessage</b></p>

	圖形
	
Twitter	功能
	<p>Twitter 元件可進行 Twitter 相關動作。</p> <p>Twitter 元件為一非可視元件，它可讓使用者與 Twitter 進行通訊。主要方法有搜尋 Twitter (SearchTwitter) 與登入 (Authorize)。一旦使用者登入成功且 IsAuthorized 事件也確認登入成功，您就可以使用以下方法：</p> <ul style="list-style-type: none"> <li>• <b>SetStatus</b>：設定使用者本身狀態。</li> <li>• <b>DirectMessage</b>：對指定使用者發送訊息。</li> <li>• <b>RequestDirectMessages</b>：接收最新的訊息。</li> <li>• <b>Follower</b>：追蹤指定使用者。</li> </ul> <ul style="list-style-type: none"> <li>• <b>StopFollowing</b>：停止追蹤指定使用者。</li> <li>• <b>RequestFollowers</b>：取得誰正在追蹤我這份清單。</li> <li>• <b>RequestFriendTimeline</b>：取得追蹤使用者的最新消息。</li> <li>• <b>RequestMentions</b>：取得有提到使用者的最新消息。</li> </ul> <p>一般來說使用上述方法時，您都可以從對應的事件來取得結果。例如使用 RequestFollowers 方法時，當追蹤者清單實際存在時就會呼叫 FollowersReceived 事件。這個動作要用掉一點時間，因為資料發送接收都要透過網路。另外當您的裝置沒有連上網際網路或是 Twitter 網站掛掉時，這時可能無法收到任何結果。</p> <p>登入 Twitter 是透過 Twitter API (dev.twitter.com/pages/auth) 所規定的 OAuth 協定來完成。如果您的 App Inventor 程式需要以合法使用者的身分進行 Twitter 相關操作，則第一件是要先使用 Authorize 方法。這個方法會顯示 Twitter 登入頁面，使用者只要輸入正確的帳號密碼就可以了，接著會程式中的 Twitter 元件就會收到一個認證 (credential)。一旦 Twitter 元件收到這個認證之後就會呼叫 IsAuthorized 事件，這時應用程式就可以執行 Twitter 相關的操作了。為一個不需要認證就能執行的 Twitter 元件下的方法就是 SearchTwitter。一般來說，應用程式都會保留使用者的登入認證，所以使用者不需要每次都執行登入動作 (除非使用者自行登出了 Twitter)。重新安裝這個應用程式將會使用 DeAuthorize 方法來清除認證。使用者之後可以從 Twitter 網站上的 Setting 頁面來取消對於應用程式的 Twitter 授權。若要檢查您的應用程式是否已取得認證，請使用 CheckAuthorized 方法。</p>



請注意	
Twitter	<p>請注意：OAuth 協定絕大部分的動作都已經被隱藏在 Twitter 元件中，您不需要瞭解它們也能夠順利使用 Twitter 元件。但是，對於任何想要在應用程式中使用 Twitter 元件的開發者，都必須提供 Consumer key 與 Consumer secret 這兩項必要資訊。您需要在 <a href="https://twitter.com/oauth_clients/new">twitter.com/oauth_clients/new</a> 頁面中註冊您的應用程式，註冊完成後就可以得到這兩筆資料（格式為字串），如此才能讓 Twitter 成功辨認您的應用程式是否有權限可以進行相關操作。在註冊頁中，您需要提供以下資訊：</p> <p><b>Application name</b> 您應用程式的名稱，名稱應為唯一（unique），當應用程式名稱並非唯一時系統會發出警告。當應用程式使用者要求要登入 Twitter 時，系統會將這個名稱作為呼叫 Authorize 方法的結果來回傳。</p> <p><b>Description</b> 您的應用程式的說明。</p> <p><b>Application website</b> 使用者可找到更多有關這個應用程式資訊或是下載本應用程式的網站（如果您有建置這樣一個網站的話）。或者您可以填入其他有用的網站。 請注意本屬性為必填。</p>
	<p><b>Application type</b> 本屬性必須設定為 Browser（瀏覽器）。</p> <p><b>Callback URL</b> 本屬性必須為一個有效的 URL，Twitter 元件會將其自動調整為適當的內容。</p> <p><b>Default access type</b> 請將本屬性設定為 read/write（讀 / 寫），代表應用程式可在 Twitter 中讀取或寫入資料。</p> <p>如果覺得累的話，其他欄位就可以不用填了。當您的應用程式時註冊成功時，您會在頁面上看到 Consumer key and Consumer secret 這兩個專屬於本應用程式的字串。您可以將它們複製到應用程式中對應的 Twitter 元件屬性裡。如果您日後想更改這些設定，只要透過瀏覽器登入 Twitter 後，找到 <a href="https://twitter.com/apps">twitter.com/apps</a> 頁面來找到您所註冊的應用程式，就可以進行修改了。</p>

屬性	
Twitter	<p><b>ConsumerKey</b></p> <p>從 twitter.com/oauth_clients/new 所取得的 ConsumerKey 字串，用以確認應用程式的身分。</p> 
	<p><b>ConsumerSecret</b></p> <p>從twitter.com/oauth_clients/new所取得的ConsumerSecret字串，用以確認應用程式的身分。</p> 
	<p><b>DirectMessages</b></p> <p>使用者在 Twitter 上所接收到的訊息。</p> 
	<p><b>Followers</b></p> <p>追蹤者清單。</p> 
	<p><b>FriendTimeline</b></p> <p>使用者的 Twitter 消息時間軸。</p> 
	<p><b>Mentions</b></p> <p>有提到使用者的最新消息，以清單格式回傳。</p> 
	<p><b>SearchResults</b></p> <p>執行一次 Twitter 搜索後的結果。</p> 



**Username**  
已授權的使用者名稱，如果回傳值為空代表沒有這一位使用者。

`Twitter1.Username`

事件

**DirectMessagesReceived(list messages)**  
當所有藉由 RequestDirectMessages 方法所查詢的訊息都收到時，呼叫本事件。

when **Twitter1.DirectMessagesReceived** messages name messages

do

**FollowersReceived(list followers)**  
當所有藉由 RequestFollowers 方法所查詢的追蹤者名單都收到時，呼叫本事件。

when **Twitter1.FollowersReceived** followers name followers

do

**FriendTimelineReceived(list user-messages-list)**  
當所有藉由 RequestFriendTimeline 方法所查詢的時間軸資訊都收到時，呼叫本事件。  
回傳清單的每一個元素都是一個包含兩個元素的清單，其中第一個元素為追蹤者名稱 (username)，第二個則是該使用者的狀態 (status)。

when **Twitter1.FriendTimelineReceived** timeline name timeline

do

## Twitter

**IsAuthorized**

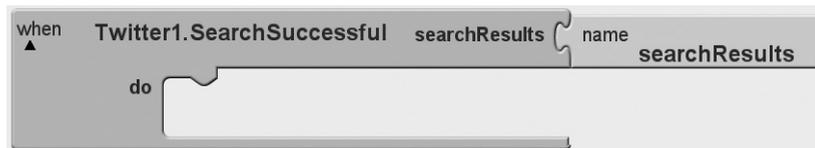
當應用程式使用 `Authorize` 方法且成功登入時呼叫本事件。或者在登入之後使用 `CheckAuthorized` 方法也會呼叫本事件。本事件成功呼叫後即可使用 Twitter 元件中的所有方法。

**MentionsReceived(list mentions)**

當所有藉由 `RequestMentions` 方法所查詢之提到使用者最新消息 (mention) 時，呼叫本事件。

**SearchSuccessful(list searchResults)**

當 `SearchTwitter` 方法成功完成一次搜尋時，呼叫本事件。



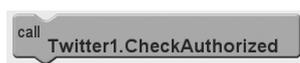
## 方法

**Authorize**

本方法會顯示 Twitter 的登入頁面，使用者可由此登入。當使用者成功登入之後會呼叫 `IsAuthorize` 事件。

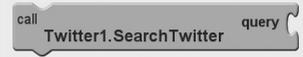
**CheckAuthorized**

回傳使用者是否已登入。如果已登入，也會呼叫 `IsAuthorize` 事件。





Twitter	<p><b>DeAuthorize</b> 讓使用者從應用程式中登出。使用者需要再次登入才能使用 Twitter 元件的方法，例如 SearchTwitter 方法。</p> <p>call <b>Twitter1.DeAuthorize</b></p>
	<p><b>DirectMessage(Text user, Text message)</b> 對指定使用者 user 發送訊息 message。</p> <p>call <b>Twitter1.DirectMessage</b> user message</p>
	<p><b>Follow(Text user)</b> 追蹤指定使用者 user。</p> <p>call <b>Twitter1.Follow</b> user</p>
	<p><b>RequestDirectMessages</b> 取得最新的訊息。</p> <p>call <b>Twitter1.RequestDirectMessages</b></p>
	<p><b>RequestFollower</b> 取得指定使用者的追隨者清單。</p> <p>call <b>Twitter1.RequestFollowers</b></p>
	<p><b>RequestFriendTimeline</b> 取得您在時間軸上最新的 20 則消息。回傳清單的每一個元素都是一個包含兩個元素的清單，其中第一個元素為發佈者名稱 (username)，第二個則是消息內容 (status message)。</p> <p>call <b>Twitter1.RequestFriendTimeline</b></p>
	<p><b>RequestMentions</b> 取得有提到使用者的消息。</p> <p>call <b>Twitter1.RequestMentions</b></p>

Twitter	<p><b>SearchTwitter(Text query)</b> 將指定文字 query 在 Twitter 中進行一次搜索。</p> 
	<p><b>SetStatus(Text s)</b> 將指定文字 status 發佈為個人最新消息。</p> 
	<p><b>StopFollowing(Text user)</b> 停止追蹤指定使用者 user。</p> 

## B-5 Sensor 感應器元件

AccelerometerSensor  
LocationSensor  
OrientationSensor

	圖形
	屬性
	 
	功能
Accelerometer-Sensor 加速度感應器	<p>AccelerometerSensor 元件可回傳 Android 裝置上的加速度感應器狀態，並可偵測裝置三個軸向的晃動狀況，偵測單位為 <math>m/s^2</math>。如果裝置面朝上水平靜置時，Z 軸加速度值約為 9.8（受地心引力影響）。</p> <ul style="list-style-type: none"> <li>• X 軸：正面時，向右傾斜，左側值增大，右側值減小，反之為負。</li> <li>• Y 軸：正面時，向下傾斜，上方值增大，下方值減小，反之為負。</li> <li>• Z 軸：當顯示朝上為正，反之為負。</li> </ul>



Accelerometer-Sensor  
加速度感應器

**Available**  
說明當下的 Android 裝置是否具備加速度感應器硬體。

**AccelerometerSensor1.Available**

---

**Enabled**  
啟動加速度感應器。

set AccelerometerSensor1.Enabled to AccelerometerSensor1.Enabled

---

**XAccel**  
加速度感應器之 X 軸變化量。

**AccelerometerSensor1.XAccel**

---

**YAccel**  
加速度感應器之 Y 軸變化量。

**AccelerometerSensor1.YAccel**

---

**ZAccel**  
加速度感應器之 Z 軸變化量。

**AccelerometerSensor1.ZAccel**

---

**事件**

**AccelerationChanged ( number xAccel, number yAccel, number zAccel )**  
當加速度感應器值改變時呼叫本事件。

when AccelerometerSensor1.AccelerationChanged

xAccel	name	xAccel
yAccel	name	yAccel
zAccel	name	zAccel

do

<p>Accelerometer-Sensor 加速度感應器</p>	<p><b>Shaking()</b></p> <p>當 Android 裝置時正被搖動時會持續呼叫本事件。</p> 
<p>Location-Sensor 位置感應器</p>	<div style="background-color: #cccccc; text-align: center; padding: 2px;">圖形</div> <div style="display: flex; justify-content: space-around; align-items: center; padding: 10px;">  <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0;">             component <b>LocationSensor1</b> </div> </div> <div style="background-color: #cccccc; text-align: center; padding: 2px;">功能</div> <p>Location Sensor 元件可提供 Android 裝置現在的位置，第一優先是使用 Android 裝置上的 GPS，接著是其他定位方法例如行動基地臺或是無線網路來定位。</p> <p>LocationSensor為一非可視元件，可提供有關位置的資訊，包括經度、緯度、海拔高度（某些裝置可能不支援）和地址。它還支援地理編碼，將指定地址（不一定要是裝置當下的位置）轉換為經度和緯度，分別使用 LatitudeFromAddress()與LongitudeFromAddress()等指令。</p> <p>正常使用 LocationSensor 之前，必須將其 Enabled 屬性設為真，另外裝置本身必須能夠通過 GPS 衛星或上述其他方法進行位置感測。</p> <div style="background-color: #cccccc; text-align: center; padding: 2px;">屬性</div> <div style="padding: 10px;"> <p><b>Accuracy</b></p> <p>回傳 Android 裝置的精度等級，單位為公尺。</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin-top: 5px;">             LocationSensor1.Accuracy           </div> </div> <div style="padding: 10px;"> <p><b>Altitude</b></p> <p>回傳 Android 裝置海拔的高度，視硬體支援程度而定。</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin-top: 5px;">             LocationSensor1.Altitude           </div> </div> <div style="padding: 10px;"> <p><b>AvailableProviders</b></p> <p>回傳可用的服務提供者清單，例如像是 GPS 或網路。</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin-top: 5px;">             LocationSensor1.AvailableProviders           </div> </div>



Location-Sensor  
位置感應器

**CurrentAddress**

回傳 Android 裝置所在地的地址。

```
LocationSensor1.CurrentAddress
```

**Enabled**

本項需設為真，才可取得位置資訊。

```
set LocationSensor1.Enabled to LocationSensor1.Enabled
```

**HasAccuracy**

本項如果為真，代表本 Android 裝置可以回傳精度。

```
LocationSensor1.HasAccuracy
```

**HasAltitude**

本項如果為真，代表本 Android 裝置可以回傳海拔高度。

```
LocationSensor1.HasAltitude
```

**HasLongitudeLatitude**

本項如果為真，代表本 Android 裝置可以回傳經度和緯度。

```
LocationSensor1.HasLongitudeLatitude
```

**Latitude**

Android 裝置所在位置的緯度。

```
LocationSensor1.Latitude
```

**Longitude**

Android 裝置所在位置的經度。

```
LocationSensor1.Longitude
```

Location-Sensor  
位置感應器

**ProviderLocked**

Android 裝置將鎖定現在的服務提供者。



**ProviderName**

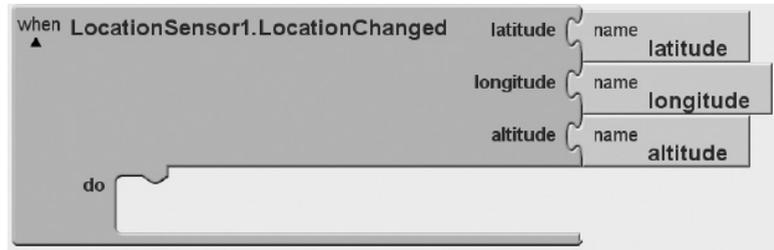
目前服務提供者名稱。



事件

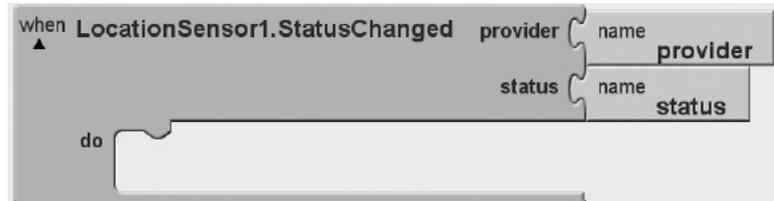
**LocationChanged ( number latitude, number longitude, number altitude )**

Android 裝置位置改變時呼叫本事件。



**StatusChanged ( text provider, text status )**

Android 裝置的服務提供者改變時呼叫本事件。



方法

**LatitudeFromAddress ( text locationName )**

回傳指定地址的緯度。





<p>Location-Sensor 位置感應器</p>	<p><b>LongitudeFromAddress</b> (text locationName)</p> <p>回傳指定地址的經度。</p> 
<p>Orientation-Sensor 姿態感應器</p>	<div style="background-color: #cccccc; text-align: center; padding: 5px;">圖形</div> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <div style="background-color: #cccccc; text-align: center; padding: 5px;">功能</div> <p>OrientationSensor 元件可用來偵測 Android 裝置在空間的定位 / 姿態。</p> <p>OrientationSensor 元件為一非可視元件，可回傳以下三種資料，皆以度為單位：</p> <ul style="list-style-type: none"> <li>• <b>Roll 滾翻</b>：裝置水平放置時為 0 度；裝置向右侧傾斜（左側較高）時會漸增到 90 度，反之則漸減到 90 度。</li> <li>• <b>Pitch 俯仰</b>：裝置水平放置時為 0 度；裝置向前端傾斜（尾巴較高）時會漸增到 90 度，整個翻面則為 180 度。反之則漸減到 90 度，反向翻面則為 -180 度。</li> <li>• <b>Azimuth 方位</b>：當裝置朝向北方時為 0 度，東方為 90 度，南方為 180 度，西方為 270 度。以上數值皆假定裝置本身並不移動。</li> </ul> <p><b>Available 可用</b></p> <p>回傳本台 Android 裝置上是否有姿態感應器可用。</p>  <p><b>Enable 啟動</b></p> <p>本項需設為真，才可使用姿態感應器。</p>  <p><b>Azimuth 方位</b></p> <p>回傳裝置的方位角。</p> <p><b>Pitch 俯仰</b></p> <p>回傳裝置的俯仰角。</p> 

Orientation-Sensor  
姿態感應器

**Roll 滾翻**  
回傳裝置的滾翻角。

**OrientationSensor1.Roll**

**Magnitude 傾斜程度**  
回傳一個介於 0 到 1 之間的小數來代表目前裝置的傾斜程度，您可以想像一個小球在裝置上的滾動速度來表示這個數值的變化情況。

**OrientationSensor1.Magnitude**

**Angle 角度**  
回傳一個角度值代表目前裝置的傾斜角，也就是說如果我們在裝置上放一個小球，Angle 可以告訴我們球該往那邊滾動。

**OrientationSensor1.Angle**

事件

OrientationChanged ( number azimuth, number pitch, number roll )  
姿態改變時呼叫本事件。

when **OrientationSensor1.OrientationChanged**

yaw name yaw

pitch name pitch

roll name roll

do

## B-6 Screen Arrangement 螢幕排列元件

HorizontalArrangement

TableArrangement

VerticalArrangement



Horizontal-Arrangement 橫向排列	<b>圖形</b>
	 <div style="border: 1px solid gray; padding: 5px; display: inline-block; margin-left: 20px;">             component <b>HorizontalArrangement1</b> </div>
	<b>功能</b>
	<p>HorizontalArrangement 元件可將多個元件從左到右橫向排列。</p> <p>本元件只用來排版，請將您想要橫向排列的元件放入其中即可。如果您希望某個元件排在另一個元件上方，請改用 VerticalArrangement。</p> <p>在 HorizontalArrangement 中的元件是沿水平方向排列，並垂直對準中心。</p> <p>有關 HorizontalArrangement 的尺寸設定，請看以下說明：</p> <ol style="list-style-type: none"> <li>1. 如果 HorizontalArrangement 的 Height 屬性為 Automatic，則其實際高度將由它所包含元件中最高的那一個來決定，但該元件的 Height 屬性不可設為 Fill Parent。</li> <li>2. 如果 HorizontalArrangement 的 Height 屬性為 Automatic，但則它所包含元件 Height 屬性為 Fill Parent，則其實際高度將以所包含元件的高度來自動決定。</li> <li>3. 如果 HorizontalArrangement 的 Height 屬性為 Automatic 且未包含任何元件，則其高度為 100 像素。</li> <li>4. 如果 HorizontalArrangement 的 Width 屬性為 Automatic，則其實際寬度將由所有元件的總寬度來決定。</li> <li>5. 如果 HorizontalArrangement 的 Width 屬性為 Automatic，則所包含元件的 Width 屬性則便設為 Fill Parent，但還是會當作 Automatic 來處理。</li> <li>6. 如果 HorizontalArrangement 的 Width 屬性為 Fill Parent 或指定像素，則所包含元件的 Width 屬性將被自動設為 Fill Parent，並平均排列不會重疊。</li> </ol>
<b>屬性</b>	
	<p><b>Visible</b></p> <p>本項需設為真，才能在螢幕上看到本元件。</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;">             set <b>HorizontalArrangement1.Visible</b> to <b>HorizontalArrangement1.Visible</b> </div>
	<p><b>Height</b></p> <p>元件高度 (y 軸像素)。</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;">             set <b>HorizontalArrangement1.Height</b> to <b>HorizontalArrangement1.Height</b> </div>
	<p><b>Width</b></p> <p>元件寬度 (x 軸像素)。</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;">             set <b>HorizontalArrangement1.Width</b> to <b>HorizontalArrangement1.Width</b> </div>

	<b>圖形</b>
	
	<b>功能</b>
	<p><b>TableArrangement</b> 元件可將多個元件以表格方式排列。</p> <p>本元件只用來排版，請將您想要以表格排列的元件放入其中即可。</p> <p>包含於 <b>TableArrangement</b> 中的所有元件將被放在不同行與列的格子中，每個格子只有一個元件能被看見。如果多個元件放在同一個格子中，則只有最後被放進去的元件才能被看到。</p> <p>每一行中的元件皆彼此垂直對準中心。</p> <p>在該列中最寬的元件確定列的寬度。當計算列的寬度時，<b>Width</b> 屬性為 <b>Fill Parent</b> 的元件將由系統自動分配其寬度。但請注意，每一個元件都會自動填滿該行寬度。</p> <p>在該行中最高的元件確定列的高度，但這些元件的 <b>Height</b> 屬性不可為 <b>Fill Parent</b>。如果該列所包含元件的 <b>Height</b> 屬性皆為 <b>Fill Parent</b>，則該列的實際高度將以所包含元件的高度來自動決定。</p>
<b>屬性</b>	
Table- Arrangement 橫向排列	<p><b>Visible</b></p> <p>本項需設為真，才能在螢幕上看到本元件。</p> 
	<p><b>Rows (number-of-rows) 行數</b></p> <p>設定列表中的行數</p>
	<p><b>Columns (number-of-columns) 列數</b></p> <p>設定列表中的列數。</p>
	<p><b>Height</b></p> <p>元件高度 (y 軸像素)。</p> 
	<p><b>Width</b></p> <p>元件寬度 (x 軸像素)。</p> 

Vertical- Arrangement 橫向排列	圖形
	功能
	屬性



**VerticalArrangement** 元件可將多個元件從上到下排列，並對齊左邊。  
本元件只用來排版，請將您想要縱向排列的元件放入其中即可。第一個子元件將位於最上方，第二、第三就依序往下垂直排列。如果您想要將元件左右排再一起，請使用 **HorizontalArrangement**。

- 在 **VerticalArrangement** 中的所有元件都是垂直排列並對齊左邊。  
有關 **VerticalArrangement** 的尺寸設定，請看以下說明：
1. 如果 **VerticalArrangement** 的 **Width** 屬性為 **Automatic**，則其實際寬度將由其下最寬的元件來決定，但該最寬元件的 **Width** 屬性不可為 **Fill Parent**。
  2. 如果 **VerticalArrangement** 的 **Width** 屬性為 **Automatic**，且其下的元件 **Width** 屬性皆為 **Fill Parent**，則其實際寬度將以所包含元件的寬度來自動決定。
  3. 如果 **VerticalArrangement** 的 **Width** 屬性為 **Automatic** 且未包含任何元件，則其高度為 100 像素，則寬度將為 100。
  4. 如果 **VerticalArrangement** 的 **Height** 屬性為 **Automatic**，則其實際高度將由所有元件總高度來決定。
  5. 如果 **VerticalArrangement** 的 **Height** 屬性為 **Automatic**，則所包含元件的 **Height** 屬性則便設為 **Fill Parent**，但還是會當作 **Automatic** 來處理。
  6. 如果 **VerticalArrangement** 的 **Height** 屬性為 **Fill Parent** 或指定像素，則所包含元件的 **Height** 屬性將被自動設為 **Fill Parent**，並平均排列不會重疊。

**Visible**  
本項需設為真，才能在螢幕上看到本元件。

**Height**  
元件高度 (y 軸像素)。

**Width**  
元件寬度 (x 軸像素)。

## B-7 LEGO MINDSTORMS 樂高機器人控制元件

NxtDirectCommands

NxtColorSensor

NxtLightSensor

NxtSoundSensor

NxtTouchSensor

NxtUltrasonicSensor

NxtDrive

本系列元件可透過藍牙對樂高 NXT 智慧型機器人套件進行各種控制。LEGO 和 MINDSTORMS 是樂高集團的註冊商標。

請注意！要使用 App Inventor 來控制樂高 NXT 機器人時，都需要在 Designer 頁面中新增一個 BluetoothClient 元件，它是用來處理手機與機器人之間的藍牙通訊。注意一個 BluetoothClient 元件只能與一臺機器人連接，所以如果您要在同一個 App Inventor 程式中與兩臺以上的機器人進行藍牙連線時，就需要根據機器人數量加入對應的 BluetoothClient 元件，BluetoothClient 元件位於 Not ready for prime time 選單中。

以下是進行一臺或多臺 NXT 機器人之藍牙通訊之前所需的步驟：

1. 在 Palette 清單中的 Not ready for prime time 選單中，將一個 BluetoothClient 元件新增到 Viewer 中，它會自動取名為 BluetoothClient1。
2. 在 Palette 清單中的 LEGO MINDSTORMS 選單中，並將一個 NxtDirectCommands 元件新增到 Viewer 中。並將它的 BluetoothClient 欄位指定為剛剛新增的 BluetoothClient1。



3. 在元件面板中，按一下「LEGO」。
4. 請拖動其中一個元件，例如 NxtDirectCommands，並將其放到 Viewer 中。
5. 在屬性區中，按一下 BluetoothClient (目前「None...」) 屬性，會出現應用程式中所有的 BluetoothClient 元件。
6. 點擊 BluetoothClient1，然後按一下確定。
7. 如果需要，添加另一個 Nxt 元件，例如 NxtColorSensor，並重複以上步驟來設定 BluetoothClient 屬性。

圖形	
<b>NxtDirectCommands</b>	
<b>component</b> NxtDirectCommands1	
功能	
NxtDirectCommands 元件可藉由低階介面與樂高 NXT 機器人進行通訊，並發送 NXT Direct Command。NXT Direct Command 是樂高公司對於 NXT 機器人所提供的特殊藍牙通訊規格，不需要編寫機器人端程式就可透過 NXT Direct Command 直接控制樂高 NXT 機器人。	
屬性	
<b>BluetoothClient</b>	欲進行通訊的 BluetoothClient 元件，本項只能在 Designer 中設定。

	方法
Nxt- Direct- Commands	<p><b>DeleteFile(text fileName)</b> 刪除機器人上的檔案。</p> <p>call <b>NxtDirectCommands1.DeleteFile</b> fileName</p>
	<p><b>DownloadFile(text source, text destination)</b> 將檔案下載到機器人。</p> <p>call <b>NxtDirectCommands1.DownloadFile</b> source destination</p>
	<p><b>GetBatteryLevel</b> 機器人目前的電池電量，單位為毫伏特。</p> <p>call <b>NxtDirectCommands1.GetBatteryLevel</b></p>
	<p><b>GetBrickName</b> 取得 NXT 主機名稱。</p> <p>call <b>NxtDirectCommands1.GetBrickName</b></p>
	<p><b>GetCurrentProgramName</b> 取得機器人上現在運行的程式名稱。</p> <p>call <b>NxtDirectCommands1.GetCurrentProgramName</b></p>



Nxt-  
Direct-  
Commands

### GetFirmwareVersion

以清單格式取得韌體（第一個項目）與通訊協定版本（第二個項目）。

```
call  
NxtDirectCommands1.GetFirmwareVersion
```

### GetInputValues(text sensorPortLetter)

讀取機器人指定輸入端的資訊，假設我們已藉由 SetInputMode 設定了感應器類型（sensor type）。

```
call  
NxtDirectCommands1.GetInputValues sensorPortLetter
```

### GetOutputState(text motorPortLetter)

讀取機器人指定輸出端的馬達狀態。

```
call  
NxtDirectCommands1.GetOutputState motorPortLetter
```

### KeepAlive

讓機器人保持開機，並每毫秒回傳一次現在的休眠時間限制。

```
call  
NxtDirectCommands1.KeepAlive
```

### ListFiles(text wildcard)

使用者設定條件（wildcard）之後，以清單回傳機器人中符合的檔案。

```
call  
NxtDirectCommands1.ListFiles wildcard
```

Nxt-  
Direct-  
Commands

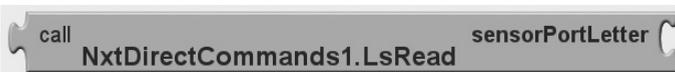
**LsGetStatus(text sensorPortLetter)**

回傳可供讀取的位元組數。Ls 意指 Low speed 低速通訊，NXT 主機支援 I<sup>2</sup>C 的數位通訊方法，可連接任何支援 I2C 介面的 I/O 裝置。



**LsRead(text sensorPortLetter)**

從機器人指定輸入端讀取低速資料（值皆為正），假設我們已藉由 SetInputMode 設定了感應器類型（sensor type）。



**LsWrite(text sensorPortLetter, list list, number rxDataLength)**

對機器人指定輸入端讀取低速資料，假設我們已藉由 SetInputMode 設定了感應器類型（sensor type）。



**MessageRead(number mailbox)**

從機器人上的指定信箱讀取資料，信箱編號由 1 到 10。



**MessageWrite(number mailbox, text message)**

對機器人上的指定信箱寫入資料，信箱編號由 1 到 10。





Nxt-  
Direct-  
Commands

**PlaySoundFile(text fileName)**

播放機器人上的音效檔，附檔名為 .rso。

```
call
  NxtDirectCommands1.PlaySoundFile  fileName
```

**PlayTone(number frequencyHz, number durationMs)**

讓機器人發出指定時間長度（durationMs，單位為毫秒）的音高（frequencyHz，赫茲）。

```
call
  NxtDirectCommands1.PlayTone  frequencyHz  durationMs
```

**ResetInputScaledValue(text sensorPortLetter)**

重設機器人指定輸入端的正規化值（scaled value）。

```
call
  NxtDirectCommands1.ResetInputScaledValue  sensorPortLetter
```

**ResetMotorPosition(text motorPortLetter, boolean relative)**

重設馬達位置。

```
call
  NxtDirectCommands1.ResetMotorPosition  motorPortLetter  relative
```

**SetBrickName(text name)**

設定 NXT 主機名稱。

```
call
  NxtDirectCommands1.SetBrickName  name
```

**SetInputMode(text sensorPortLetter, number sensorType, number sensorMode)**

設定機器人的指定輸入端狀態。sensorPortLetter 為輸入端編號 1~4，sensorType 為感應器類型，sensorMode 為感應器回傳值格式。

```
call
  NxtDirectCommands1.SetInputMode  sensorPortLetter  sensorType  sensorMode
```

Nxt-  
Direct-  
Commands

**SetOutputState(text motorPortLetter, number power, number mode, number regulationMode, number turnRatio, number runState, number tachoLimit)**

設定機器人的指定輸出端狀態，motorPortLetter 為輸出端編號 A-C，umber power 為馬達電力範圍 -100~100，mode 為資料模式，regulationMode 為馬達控制模式，turnRatio 為轉彎百分比 -100~100，runState 為執行狀態，tachoLimit 為角度感應器上限。



**StartProgram(text programName)**

執行已下載到機器人上的程式。



**StopProgram**

停止機器人現在運行中的程式。



**StopSoundPlayback**

停止播放聲音。





Nxt-  
ColorSensor

圖形

**NxtColorSensor** 顏色感應器



component

**NxtColorSensor1**

功能

NxtColorSensor 元件可用來控制樂高 NXT 機器人上的顏色感應器。

屬性

**BluetoothClient**

用於通訊的 BluetoothClient 元件，必須在 Designer 中設定。

**SensorPort**

感應器所連接的輸入端，必須在 Designer 中設定。

**DetectColor**

設定顏色感應器要偵測顏色或是光值。設定為 true 代表應偵測顏色變化，反之則如同光感應器一樣偵測光值變化。

如果 DetectColor 屬性設定為 True，那麼 BelowRange、WithinRange、AboveRange 等事件都不會被呼叫，感應器前端也不會發光。

如果 DetectColor 屬性設定為 False，則 ColorChanged 事件不會被呼叫。

set **NxtColorSensor1.DetectColor** to **NxtColorSensor1.DetectColor**

**ColorChangedEventEnabled**

當 DetectColor 屬性設定為 true 且偵測到的顏色發生變化時，設定是否呼叫 ColorChanged 事件。

set **NxtColorSensor1.ColorChangedEventEnabled** to **NxtColorSensor1.ColorChangedEventEnabled**

Nxt-  
ColorSensor**GenerateColor**

設定顏色感應器是否會發光，只接受紅綠藍等三種顏色，這也是顏色感應器所能產生的三種顏色。當 DetectColor 屬性設定為 true 時，顏色感應器不會發光。

**BottomOfRange**

**BelowRange**、**WithinRange** 和 **AboveRange** 等事件的最小值。

**TopOfRange**

**BelowRange**、**WithinRange** 和 **AboveRange** 等事件的最大值。

Nxt-  
ColorSensor**BelowRangeEventEnabled**

決定當 DetectColor 屬性設定為 false 且光值低於 BottomOfRange 時，是否呼叫 BelowRange 事件。

**WithinRangeEventEnabled**

決定當 DetectColor 屬性設定為 false 且光值介於 BottomOfRange 與 TopOfRange 時，是否呼叫 WithinRange 事件。

**AboveRangeEventEnabled**

決定當 DetectColor 屬性設定為 false 且光值高於 TopOfRange 時，是否呼叫 TopOfRange 事件。





事件

**ColorChanged(number color)**

偵測到的顏色已改變。

如果 DetectColor 或 ColorChangedEventEnabled 屬性任一項設定為 False，ColorChanged 事件不會發生。

Nxt-  
ColorSensor

when **NxtColorSensor1.ColorChanged** color name **color**

do

**BelowRange**

光值已經低於指定範圍。

如果 DetectColor 屬性設定為 True 或 BelowRangeEventEnabled 屬性設定為 False，BelowRange 事件不會發生。

when **NxtColorSensor1.BelowRange**

do

**WithinRange**

光值介於指定範圍之間。

如果 DetectColor 屬性設定為 True 或 BelowRangeEventEnabled 屬性設定為 False，BelowRange 事件不會發生。

when **NxtColorSensor1.WithinRange**

do

**AboveRange**

光值已經高於指定範圍。

如果 DetectColor 屬性設定為 True 或 AboveRangeEventEnabled 屬性設定為 False，AboveRange 事件不會發生。

Nxt-  
ColorSensor

when **NxtColorSensor1.AboveRange**

do

## 方法

**GetColor**

回傳所偵測到的顏色，如果回傳值為 none 代表無法辨識顏色或因為 DetectColor 屬性設定為 false 所導致。

call  
NxtColorSensor1.GetColor

**GetLightLevel**

回傳光值強度，這是一個介於 0 到 1023 之間的整數，如果回傳 -1 代表無法讀取光值或因為 DetectColor 屬性設定為 true 所導致。

call  
NxtColorSensor1.GetLightLevel

## 圖形

**NxtLightSensor 光感應器**

component  
NxtLightSensor1



## 功能

NxtLightSensor 元件可用來控制樂高 NXT 機器人上的光感應器。

## 屬性

**BluetoothClient**

用於通訊的 BluetoothClient 元件，必須在 Designer 中設定。

**SensorPort**

感應器所連接的輸入端，必須在 Designer 中設定。

NxtLight-Sensor



NxtLight-Sensor	<p><b>GenerateLight</b></p> <p>光感應器前端燈泡是否發光。</p>
	<p><b>BottomOfRange</b></p> <p>BelowRange、WithinRange 和 AboveRange 等事件的最小值。</p>
	<p><b>TopOfRange</b></p> <p>BelowRange、WithinRange 和 AboveRange 等事件的最大值。</p>
	<p><b>BelowRangeEventEnabled</b></p> <p>決定當光值低於 BottomOfRange 時，是否呼叫 BelowRange 事件。</p>
	<p><b>WithinRangeEventEnabled</b></p> <p>決定當光值介於 BottomOfRange 與 TopOfRange 之間時，是否呼叫 WithinRange 事件。</p>

NxtLight-Sensor	<b>AboveRangeEventEnabled</b> 決定當亮度高於 TopOfRange 時，是否呼叫 AboveRange 事件。
	事件
	<b>BelowRange</b> 光值已經低於指定範圍。
NxtLight-Sensor	
	<b>WithinRange</b> 光值介於指定範圍之間。
NxtLight-Sensor	<b>AboveRange</b> 光值已經高於指定範圍。
	方法
NxtLight-Sensor	<b>GetLightLevel</b> 回傳光值強度，這是一個介於 0 到 1023 之間的整數，如果回傳 -1 代表無法讀取光值。

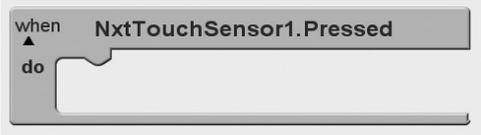


	圖形
	<p><b>NxtSoundSensor 聲音感應器</b></p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid gray; padding: 5px; margin-right: 10px;">             component  <b>NxtSoundSensor1</b> </div> <div style="border: 1px solid gray; padding: 10px; text-align: center;">  </div> </div>
	功能
NxtSound-Sensor	NxtSoundSensor 元件可用來控制樂高 NXT 機器人上的聲音感應器。
	屬性
	<p><b>BluetoothClient</b> 用於通訊的 BluetoothClient 元件，必須在 Designer 中設定。</p> <p><b>SensorPort</b> 感應器所連接的輸入端，必須在 Designer 中設定。</p>
	<p><b>BottomOfRange</b> BelowRange、WithinRange 和 AboveRange 等事件的最小值。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">             set <b>NxtSoundSensor1.BottomOfRange</b> to <b>NxtSoundSensor1.BottomOfRange</b> </div>
	<p><b>TopOfRange</b> BelowRange、WithinRange 和 AboveRange 等事件的最大值。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">             set <b>NxtSoundSensor1.TopOfRange</b> to <b>NxtSoundSensor1.TopOfRange</b> </div>
NxtSound-Sensor	<p><b>BelowRangeEventEnabled</b> 決定當音量低於 BottomOfRange 時，是否呼叫 BelowRange 事件。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">             set <b>NxtSoundSensor1.BelowRangeEventEnabled</b> to <b>NxtSoundSensor1.BelowRangeEventEnabled</b> </div>

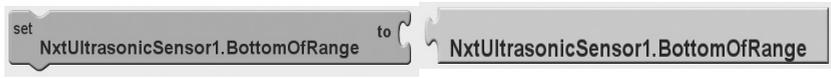
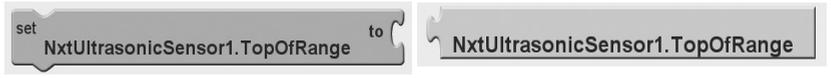
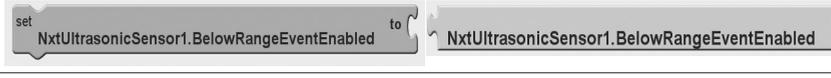
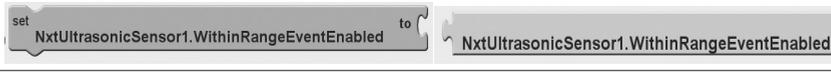
NxtSound-Sensor	<p><b>WithinRangeEventEnabled</b></p> <p>決定當音量介於 BottomOfRange 與 TopOfRange 之間時，是否呼叫 WithinRange 事件。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <span style="border: 1px solid gray; padding: 2px;">set</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.WithinRangeEventEnabled</span> <span style="border: 1px solid gray; padding: 2px;">to</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.WithinRangeEventEnabled</span> </div>	
	<p><b>AboveRangeEventEnabled</b></p> <p>決定當音量超過 TopOfRange 時，是否呼叫 AboveRange 事件。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <span style="border: 1px solid gray; padding: 2px;">set</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.AboveRangeEventEnabled</span> <span style="border: 1px solid gray; padding: 2px;">to</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.AboveRangeEventEnabled</span> </div>	
	事件	
	<p><b>BelowRange</b></p> <p>音量已經低於指定範圍</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <span style="border: 1px solid gray; padding: 2px;">when</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.BelowRange</span>  <span style="border: 1px solid gray; padding: 2px;">do</span> </div>	
	<p><b>WithinRange</b></p> <p>音量介於指定範圍之間</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <span style="border: 1px solid gray; padding: 2px;">when</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.WithinRange</span>  <span style="border: 1px solid gray; padding: 2px;">do</span> </div>	
NxtSound-Sensor	<p><b>AboveRange</b></p> <p>音量已經高於指定範圍。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <span style="border: 1px solid gray; padding: 2px;">when</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.AboveRange</span>  <span style="border: 1px solid gray; padding: 2px;">do</span> </div>	
	方法	
<p><b>GetSoundLevel</b></p> <p>回傳音量強度，這是一個介於 0 到 1023 之間的整數，如果回傳 -1 代表無法讀取音量。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <span style="border: 1px solid gray; padding: 2px;">call</span> <span style="border: 1px solid gray; padding: 2px;">NxtSoundSensor1.GetSoundLevel</span> </div>		

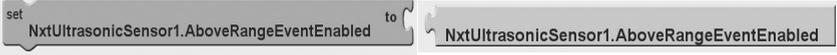
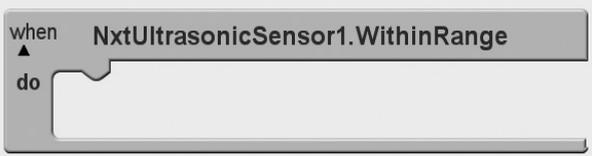
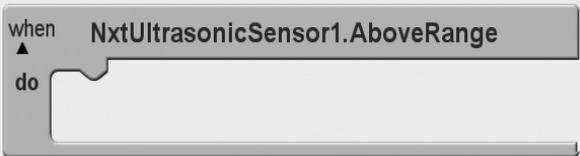


NxtTouch-Sensor	圖形
	<p><b>NxtTouchSensor 觸碰感應器</b></p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 5px; margin-right: 10px;">             component  <b>NxtTouchSensor1</b> </div> <div style="border: 1px solid #ccc; padding: 10px; text-align: center;">  </div> </div>
	功能
	<p>NxtTouchSensor 元件可用來控制樂高 NXT 機器人上的觸碰感應器。</p>
	屬性
<p><b>BluetoothClient</b> 用於通訊的 BluetoothClient 元件，必須在 Designer 中設定。</p> <p><b>SensorPort</b> 感應器所連接的輸入端，必須在 Designer 中設定。</p>	
<p><b>PressedEventEnabled</b> 設定是否可呼叫 Pressed 事件，意即觸碰感應器壓下時呼叫 Pressed 事件。</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">             set <b>NxtTouchSensor1.PressedEventEnabled</b> to <b>NxtTouchSensor1.PressedEventEnabled</b> </div>	
<p><b>ReleasedEventEnabled</b> 設定是否可呼叫 Released 事件，意即觸碰感應器壓下時呼叫 Released 事件。</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">             set <b>NxtTouchSensor1.ReleasedEventEnabled</b> to <b>NxtTouchSensor1.ReleasedEventEnabled</b> </div>	

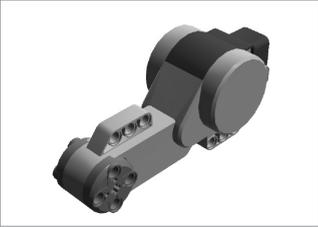
NxtTouch-Sensor	<p style="text-align: center;">事件</p> <p><b>Pressed</b></p> <p>當觸摸感應器被按下時呼叫本事件。</p>  <p>when <b>NxtTouchSensor1.Pressed</b></p> <p>do</p>
	<p><b>Released</b></p> <p>當觸摸感應器被放開時呼叫本事件</p>  <p>when <b>NxtTouchSensor1.Released</b></p> <p>do</p>
	<p style="text-align: center;">方法</p> <p><b>IsPressed</b></p> <p>如果觸摸感應器被壓下了，則回傳 true；反之回傳 false。</p>  <p>call <b>NxtTouchSensor1.IsPressed</b></p>
	<p style="text-align: center;">圖形</p> <p><b>NxtUltrasonicSensor 超音波感應器</b></p>   <p>component <b>NxtUltrasonicSensor1</b></p>



功能	
	NxtUltrasonicSensor 元件可用來控制樂高 NXT 機器人上的超音波感應器。
屬性	
NxtUltrasonic-Sensor	<p><b>BluetoothClient</b></p> <p>用於通訊的 BluetoothClient 元件，必須在 Designer 中設定。</p>
	<p><b>SensorPort</b></p> <p>感應器連接到感應器中，必須在 Designer 設定。</p>
	<p><b>BottomOfRange</b></p> <p>BelowRange、WithinRange 和 AboveRange 等事件的最小值。</p> 
	<p><b>TopOfRange</b></p> <p>BelowRange、WithinRange 和 AboveRange 等事件的最大值。</p> 
	<p><b>BelowRangeEventEnabled</b></p> <p>決定當距離低於 BottomOfRange 時，是否呼叫 BelowRange 事件。</p> 
NxtUltrasonic-Sensor	<p><b>WithinRangeEventEnabled</b></p> <p>決定當距離介於 BottomOfRange 與 TopOfRange 之間時，是否呼叫 WithinRange 事件。</p> 

NxtUltrasonic-Sensor	<p><b>AboveRangeEventEnabled</b></p> <p>決定當距離超過 TopOfRange 時，是否呼叫 AboveRange 事件。</p> 
	事件
	<p><b>BelowRange</b></p> <p>距離已經低於指定範圍。</p> 
	<p><b>WithinRange</b></p> <p>距離介於指定範圍之間。</p> 
NxtUltrasonic-Sensor	<p><b>AboveRange</b></p> <p>距離已經高於指定範圍，</p> 
	方法
<p><b>GetDistance</b></p> <p>回傳距離，單位為公分，這是一個介於 0 到 254 之間的整數，如果回傳 -1 代表無法判斷距離。</p> 	



NxtDrive-Sensor	圖形
	<p><b>NxtDriveNxt 馬達</b></p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid gray; padding: 5px; margin-right: 10px;">             component  <b>NxtDrive1</b> </div>  </div>
	功能
	<p>NxtDrive 元件可用來控制樂高 NXT 機器人上的馬達，進而控制機器人前進、後退或轉彎，或控制單顆馬達動作。</p>
	屬性
	<p><b>BluetoothClient</b> 用於通訊的 BluetoothClient 元件，必須在 Designer 中設定。</p> <p><b>DriveMotors</b> 所要控制的馬達，可輸入 A、B、C、AC、BC、AB 與 ABC 等參數。</p> <p><b>WheelDiameter</b> 裝於馬達上的輪胎直徑，單位為公分。</p> <p><b>StopBeforeDisconnect</b> 設定是否在斷線之前先把馬達停下來。</p>
	方法
<p><b>MoveForwardIndefinitely(number power)</b> 讓機器人持續以指定電力 power 前進，電力範圍為 -100~100，請注意此處電力如輸入負數，會使馬達反轉（機器人後退）。電力為 0 時馬達靜止。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">             call <b>NxtDrive1.MoveForwardIndefinitely</b> power           </div>	

NxtDrive-  
Sensor**MoveForward(number power, number distance)**

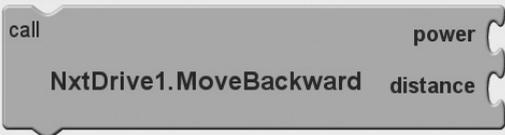
讓機器人以指定電力 **power** 前進指定距離 **distance**，距離是由 **WheelDiameter** 屬性計算求得馬達每轉 1 度時的實際前進長度。

**MoveBackwardIndefinitely(number power)**

讓機器人持續以指定電力 **power** 後退，電力範圍為 -100~100，請注意此處電力如輸入負數，會使馬達反轉（機器人前進）。電力為 0 時馬達靜止。

**MoveBackward(number power, number distance)**

讓機器人以指定電力 **power** 後退指定距離 **distance**，距離是由 **WheelDiameter** 屬性計算求得馬達每轉 1 度時的實際前進長度。

**Stop**

所有馬達停止轉動。





## B-8 Other Components 其他元件

ActivityStarter

BarcodeScanner

BluetoothClient

BluetoothServer

Notifier

SpeechRecognizer

TextToSpeech

TinyWebDB

Web

Activity- Starter	圖形
	<p><b>ActivityStarter</b></p> 
	功能
<p>ActivityStarter 元件可以讓您的應用程式呼叫另一項活動 (Activity)。透過設定 ActivityStarter 的屬性，我們就能正確地與它溝通，這包含 action 與 activity 等 class，詳細說明與範例請參考附錄 C&lt;App Inventor 小秘訣 &gt;。</p>	

可由 ActivityStarter 啟動的活動包括：

1. 啟動另一個 App Inventor 應用程式：首先確定您所要啟動類的應用程式的 class，請下載它原始碼並解壓縮，找到一個名為「youngandroidproject/project.properties」檔。第一行將由「main =」開始，後面跟著 class 名稱，例如：

```
main=com.gmail.Bitdiddle.Ben.HelloPurr.Screen1
```

若要使您的 ActivityStarter 元件能夠呼叫這個程式，請如下設定：

- 在 ActivityPackage 屬性中填入 class 名稱，但最後一個元件名稱不要放，例如 com.gmail.Bitdiddle.Ben.HelloPurr。
  - 在 ActivityClass 屬性中填入完整的 class 名稱，例如 com.gmail.Bitdiddle.Ben.HelloPurr.Screen1。
2. 啟動內建於 Android 作業系統的活動：如使用相機或執行一次網路搜尋。您可以藉由以下設定來啟動相機：
    - Action: android.intent.action.MAIN
    - ActivityPackage: com.android.camera
    - ActivityClass: com.android.camera.Camera
  3. 執行網路搜尋：假定您要搜尋「vampire」這個詞，請依循以下設定：
    - Action: android.intent.action.WEB\_SEARCH
    - ExtraKey: query
    - ExtraValue: vampire
    - ActivityPackage: com.google.android.providers.enhancedgooglesearch
    - ActivityClass: com.google.android.providers.enhancedgooglesearch.Launcher



4. 打開瀏覽器並到指定網頁：如果您想去的網站是「www.facebook.com」，請依循以下設定：

- Action: android.intent.action.VIEW
- DataUri: http://www.facebook.com

您也可以藉由 ActivityStarter 來啟動安裝在您 Android 裝置上的第三方程式，但需要提供正確的 intent 呼叫它們。您也可以呼叫會產生字串結果的活動，並將這些結果拉回到您的應用程式當中。請注意擷取資料的方式與該程式實作的方法有關。

屬性

Activity-Starter

Action

欲使用 ActivityStarter 元件啟動的 activity。

set ActivityStarter 1.Action to ActivityStarter1.Action

ActivityClass

欲呼叫 activity 之 class 名稱。

set ActivityStarter 1.ActivityClass to ActivityStarter1.ActivityClass

ActivityPackage

欲呼叫 activity 的 Package 名稱。

set ActivityStarter 1.ActivityPackage to ActivityStarter1.ActivityPackage

DataUri

傳送給欲呼叫 activity 的 URI (Uniform Resource Identifier)。

set ActivityStarter 1.DataUri to ActivityStarter1.DataUri

Activity-  
Starter**ExtraKey**

傳送給欲呼叫 activity 的 key 名稱。

**Extra Value**

傳送給欲呼叫 activity 的資料內容。

**Result**

所呼叫 activity 的回傳值內容。

**ResultName**

所呼叫 activity 回傳值的名稱。

**ResultType: text**

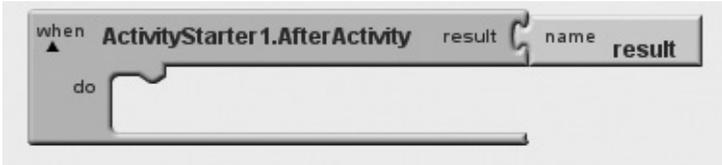
所呼叫 activity 的資料型態。

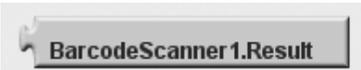
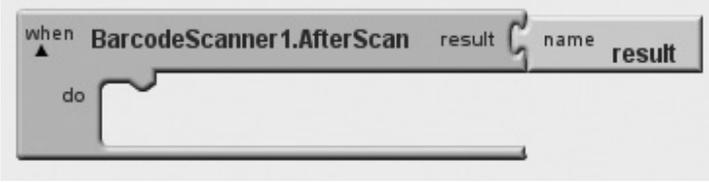
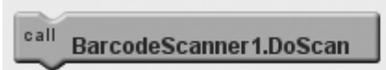
**ResultUri: text**

所呼叫 activity 傳回的 URI 或資料。

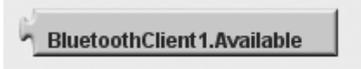




Activity- Starter	事件
	<p><b>ActivityError(text message)</b> 代表使用 ActivityStarter 時發生了錯誤。</p> <p><b>AfterActivity(text result)</b> ActivityStarter 結束後啟動本事件。</p> 
	方法
	<p><b>ResolveActivity</b> 回傳所呼叫的 activity 名稱，如果回傳一個空字串代表找不到對應的 activity。您可以在呼叫外部程式之前先使用此方法，好確保該程式已確實安裝於 Android 裝置之中。</p> 
	<p><b>StartActivity</b> 啟動欲呼叫的 activity。</p> 

Barcode-Scanner	圖形
	<p><b>BarcodeScanner 條碼掃描器</b></p> 
	功能
	<p>BarcodeScanner 為一非可視元件，它可啟動手機的照相機來讀取一維條碼或二維條碼 (QR 碼)。在使用該元件之前，必須先在您的手機上安裝 ZXing 或其他條碼掃描器軟體，此類應用程式絕大部分都是免費的。</p>
	屬性
	<p><b>Result</b></p> <p>掃描成功後回傳的字串結果。本屬性在 AfterScan 事件完成後就可存取。掃描結果也可在 Block Editor 中來取得。</p> 
事件	
<p><b>AfterScan(text result)</b></p> <p>掃描結束後呼叫本事件。</p> 	
方法	
<p><b>DoScan()</b></p> <p>開始掃描。</p> 	



Bluetooth-Client	圖形
	<p><b>BluetoothClient 藍牙用戶端</b></p> 
	功能
	<p>藍牙用戶端元件。</p>
	屬性
	<p><b>AddressesAndNames</b></p> <p>已配對藍牙裝置的名稱 / 位址清單。請注意如果您的手機從未和任何裝置進行藍牙配對，則本清單將為空。</p> 
	<p><b>Available</b></p> <p>回傳當下的 Android 裝置上是否可使用藍牙。</p> 
	<p><b>CharacterEncoding</b></p> <p>設定收發訊息時的字元編碼。</p> 
<p><b>DelimiterByte</b></p> <p>呼叫 ReceiveText、ReceiveSignedBytes、ReceiveUnsignedBytes 等函數時，當引數 numberOfBytes 為負值，須使用界定字元 (Delimiter Byte) 當資料串結尾。</p> 	
<p><b>Enabled</b></p> <p>本項需設為真，才能使用藍牙。</p> 	

Bluetooth-Client

**HighByteFirst**

2 及 4 位元組數值是否應先從最高位元開始傳送，確認通訊裝置的文件說明，應能與 App 適當通訊，Big-endian 俗稱為高位元組資料優先的排序方式。

set **BluetoothClient1.HighByteFirst** to **BluetoothClient1.HighByteFirst**

**IsConnected**

回傳是否已建立藍牙連線。

**BluetoothClient1.IsConnected**

事件

無

方法

**BytesAvailableToReceive()**

回傳在不塞車情況下的可接收位元組數（估計值）。

call **BluetoothClient1.BytesAvailableToReceive**

**Connect(text address)**

與指定位址與序列埠的藍牙裝置進行連線，如果連接成功，則回傳 true。address 參數中可在 MAC 位址後包含額外的字元。這代表您可以在不拆開 address 與 name 的情況之下，從 AddressesAndNames 屬性所回傳的清單中載送一些資料出去。

call **BluetoothClient1.Connect** address

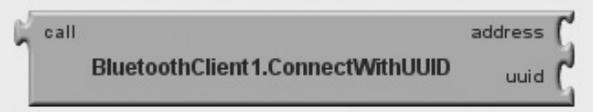


Bluetooth-Client

**ConnectWithUUID(text address, text uuid)**

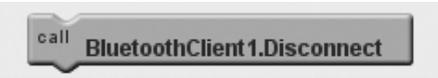
和指定位址及 UUID (Universally Unique Identifier, 通用唯一識別碼) 的藍牙裝置進行連線。如果連接成功, 則回傳 true。address 參數中可在 MAC 位址後包含額外的字元。這代表您可以在不拆開 address 與 name 的情況之下, 從 AddressesAndNames 屬性所回傳的清單中載送一些資料出去。

樂高 NXT 機器人的 UUID 皆為 00001101-0000-1000-8000-00805F9B34FB。



**Disconnect()**

中斷藍牙連線。



**IsDevicePaired(text address)**

檢查指定位址的藍牙裝置是否已配對。



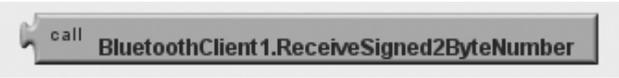
**ReceiveSigned1ByteNumber()**

從所連接的藍牙裝置接收 1 位元組長度的有正負號數值 (以下簡稱有號數)。



**ReceiveSigned2ByteNumber()**

從所連接的藍牙裝置接收 2 位元組長度的有號數。



**ReceiveSigned4ByteNumber()**

從所連接的藍牙裝置接收 4 位元組長度的有號數。



Bluetooth-Client

**ReceiveSignedBytes(number numberOfBytes)**

從所連接的藍牙裝置接收多個有號位元組值。如果 numberOfBytes 小於 0，將持續讀取直到收到一個界定字元為止。



call **BluetoothClient 1.ReceiveSignedBytes** numberOfBytes

**ReceiveText(number numberOfBytes)**

從所連接的藍牙裝置接收一個字串。如果 numberOfBytes 小於 0，將持續讀取直到收到一個界定字元為止。

從所連接的藍牙裝置接收一個字串。如果 numberOfBytes 小於 0，將持續讀取直到收到一個 delimiter byte value 為止。



call **BluetoothClient 1.ReceiveText** numberOfBytes

**ReceiveUnsigned1ByteNumber()**

從所連接的藍牙裝置接收 1 位元組長度的無號數。



call **BluetoothClient 1.ReceiveUnsigned1ByteNumber**

**number ReceiveUnsigned2ByteNumber()**

從所連接的藍牙裝置接收 2 位元組長度的無號數。



call **BluetoothClient 1.ReceiveUnsigned2ByteNumber**

**number ReceiveUnsigned4ByteNumber()**

從所連接的藍牙裝置接收 4 位元組長度的無號數。



call **BluetoothClient 1.ReceiveUnsigned4ByteNumber**

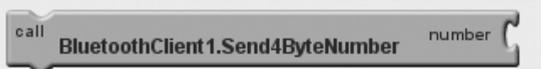
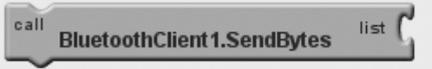
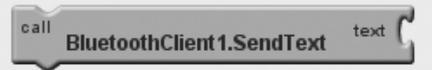
**ReceiveUnsignedBytes(number numberOfBytes)**

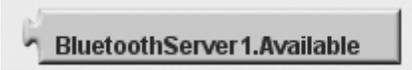
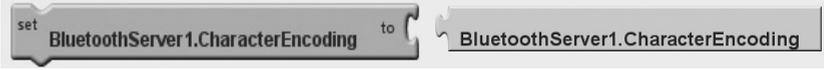
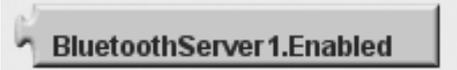
從所連接的藍牙裝置接收多個無號位元組值。如果 numberOfBytes 小於 0，將持續讀取直到收到一個界定字元為止。



call **BluetoothClient 1.ReceiveUnsignedBytes** numberOfBytes



Bluetooth-Client	<p><b>Send1ByteNumber(text number)</b></p> <p>對已連接的藍牙裝置發送 1 位元組長度的數字。</p> 
	<p><b>Send2ByteNumber(text number)</b></p> <p>對已連接的藍牙裝置發送 2 位元組長度的數字。</p> 
	<p><b>Send4ByteNumber(text number)</b></p> <p>對已連接的藍牙裝置發送 4 位元組長度的數字。</p> 
	<p><b>SendBytes(list list)</b></p> <p>對已連接的藍牙裝置發送位元組清單，即 byte array。</p> 
	<p><b>SendText(text text)</b></p> <p>對已連接的藍牙裝置發送字串。</p> 

Bluetooth- Server	圖形
	<p><b>BluetoothServer</b></p> 
	功能
	Bluetooth server component 藍牙伺服器元件。
	屬性
	<p><b>Available: Boolean</b></p> <p>回傳當下的 Android 裝置上是否可使用藍牙。</p> 
<p><b>CharacterEncoding</b></p> <p>設定收發訊息時的字元編碼。</p> 	
<p><b>DelimiterByte</b></p> <p>呼叫 ReceiveText、ReceiveSignedBytes、ReceiveUnsignedBytes 等函數時，當引數 numberOfBytes 為負值，須使用界定字元 (Delimiter Byte) 當資料串結尾。</p> 	
<p><b>Enabled</b></p> <p>本項需設為真，才能使用藍牙。</p> 	



Bluetooth-Server

**HighByteFirst: boolean**  
 2 及 4 位元組數值是否應先從最高位元開始傳送，確認通訊裝置的文件說明，應能與 app 適當通訊，Big-endian 俗稱為高位元組資料優先的排序方式。

**IsAccepting**  
 代表本 BluetoothServer 元件是否允許 BluetoothClient 的連線要求。

**IsConnected: boolean**  
 回傳是否已建立藍牙連線。

事件

**ConnectionAccepted()**  
 當藍牙連線要求已被接受時呼叫本事件。

**AcceptConnectionWithUUID(text serviceName, text uuid)**  
 接收由指定 UUID 發起的連線要求。。

**BytesAvailableToReceive()**  
 回傳在不塞車情況下的可接收位元組數（估計值）。

Bluetooth-  
Server**Disconnect()**

中斷藍牙連線。


 call **BluetoothServer 1.Disconnect**
**ReceiveSigned1ByteNumber()**

從所連接的藍牙裝置接收 1 位元組長度的有號數。


 call **BluetoothServer 1.ReceiveSigned1ByteNumber**
**ReceiveSigned2ByteNumber()**

從所連接的藍牙裝置接收 2 位元組長度的有號數。


 call **BluetoothServer 1.ReceiveSigned2ByteNumber**
**number ReceiveSigned4ByteNumber()**

從所連接的藍牙裝置接收 4 位元組長度的有號數。


 call **BluetoothServer 1.ReceiveSigned4ByteNumber**
**ReceiveSignedBytes(number numberOfBytes)**

從所連接的藍牙裝置接收多個有號位元組值。

如果 numberOfBytes 小於 0，將持續讀取直到收到一個界定字元為止。


 call **BluetoothServer 1.ReceiveSignedBytes** numberOfBytes
**ReceiveText(number numberOfBytes)**

從所連接的藍牙裝置接收一個字串。如果 numberOfBytes 小於 0，將持續讀取直到收到一個界定字元為止。


 call **BluetoothServer 1.ReceiveText** numberOfBytes
**ReceiveUnsigned1ByteNumber()**

從所連接的藍牙裝置接收 1 位元組長度的無號數。


 call **BluetoothServer 1.ReceiveUnsigned1ByteNumber**



Bluetooth-Server

**ReceiveUnsigned2ByteNumber()**

從所連接的藍牙裝置接收 2 位元組長度的無號數。

```
call BluetoothServer 1.ReceiveUnsigned2ByteNumber
```

**ReceiveUnsigned4ByteNumber()**

從所連接的藍牙裝置接收 4 位元組長度的無號數。

```
call BluetoothServer 1.ReceiveUnsigned4ByteNumber
```

**ReceiveUnsignedBytes(number numberOfBytes)**

從所連接的藍牙裝置接收多個無號位元組值。  
如果 numberOfBytes 小於 0，將持續讀取直到收到一個界定字元為止。

```
call BluetoothServer 1.ReceiveUnsignedBytes numberOfBytes
```

**Send1ByteNumber(text number)**

對已連接的藍牙裝置發送 1 位元組長度的數字。

```
call BluetoothServer 1.Send1ByteNumber number
```

**Send2ByteNumber(text number)**

對已連接的藍牙裝置發送 2 位元組長度的數字。

```
call BluetoothServer 1.Send2ByteNumber number
```

**Send4ByteNumber(text number)**

對已連接的藍牙裝置發送 4 位元組長度的數字。

```
call BluetoothServer 1.Send4ByteNumber number
```

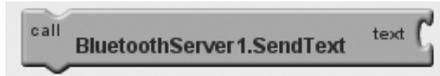
**SendBytes(list list)**

對已連接的藍牙裝置發送位元組清單，即 byte array。

```
call BluetoothServer 1.SendBytes list
```

Bluetooth-  
Server**SendText(text text)**

對已連接的藍牙裝置發送字串。

**StopAccepting()**

不再接收外部連線要求。





圖形

Notifier 通知



功能

Notifier 是不可視元件，用以顯示各種警告與系統資訊。Notifier 可以用來顯示通知與警告，並顯示系統資訊作 app 除錯之用。

方法

ShowMessageDialog(Text message, Text title, Text buttonText)

彈出訊息，直到使用者按下某按鈕回應後才會消失。參數為顯示訊息內容、標題以及按鈕標籤。



Notifier

ShowChooseDialog(Text message, Text title, Text button1Text, Text button2Text)

彈出通知，使用者必須依據內容，鍵入兩按鈕其中之一作為回應。參數為顯示的訊息本身、標題、左右兩鍵的標籤。當使用者按鈕回應之後，AfterChoosing 事件會發生。



ShowTextDialog(Text message, Text title)

彈出使用這必須輸入回應的通知，參數為訊息內文與通知名稱，在使用者輸入訊息之後，AfterTextInput 事件會出現。



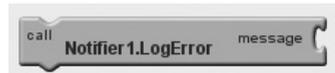
彈出臨時的通知，幾秒鐘後消失。該參數是通知的內文。



Notifier

**LogError(Text message)**

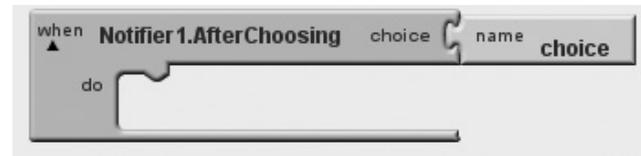
這些方法是用來除錯的，他們在 Android 裝置中各種紀錄中寫入訊息，這些紀錄可以使用 Android 的除錯工具來讀取，比如像 Android Debug Bridge (ADB)。



事件

**AfterChoosing(Text choice)**

在使用者在 ShowChooseDialog 中做出選擇後發出訊號，參數為使用者所按的按鈕代表的文字。

**AfterTextInput(Text response)**

使用者回應 ShowTextDialog 後發出信號。這個參數內容為按鍵上使用者輸入的資料。

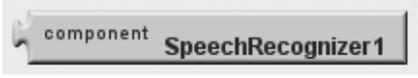




Speech-  
Recognizer

圖形

SpeechRecognizer 語音辨識器



功能

使用語音辨識器元件聽取使用者口語，並以 Android 的語音辨識功能，將語音資料轉換成文字。

屬性

Result

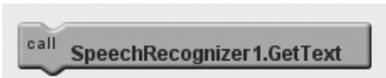
辨識器產生的最後一段文字資料。



方法

GetText()

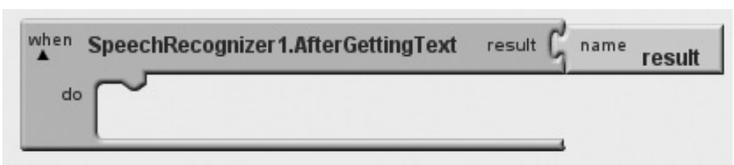
請使用者發言，並將語音資料轉換為文字資料。當有結果可用時，會發出 AfterGetting-Text 事件訊號。



事件

AfterGetting(Text result)

在辨識器產生文字資料之後發出訊號，這個參數代表產生的文字資料。



BeforeGettingText()

在辨識器呼叫之前發出訊號。



TextTo-Speech	圖形
	<p><b>TextToSpeech 語音文字</b></p> 
	功能
	<p>使您的裝置能念出文字資料。要使這項元件得以運作，您的裝置必須要裝有 Eyes-Free Project 建置的 TTS Extended Service app，在 <a href="http://code.google.com/p/eyes-free/downloads/list">http://code.google.com/p/eyes-free/downloads/list</a> 可以下載得到。文字－語音轉換元件有語音文字轉換的相關屬性設定，通常是以三個字母的代碼來表示語言以及使用區域。舉例而言，您可以區分英式與美式英語，英式英語的語言代碼為 eng，區域代碼則是 GBR；而若是美式英語，語言代碼依舊是 eng，而區域代碼則是 USA。後面會列出完整的列表。</p>
	屬性
<p><b>Country</b></p> <p>語音輸出的國家代碼。</p> 	
<p><b>Language</b></p> <p>語音輸出的語言代碼。</p> 	
<p><b>Result</b></p> <p>詳細資訊。</p> 	



TextToSpeech

方法

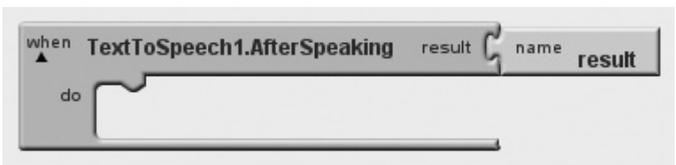
**Speak(Text message)**

發送指定文字資料。



事件

**AfterSpeaking(Text result)**



文字資料發送後的信號，該參數是依文字資料的結果產生的。

**BeforeSpeaking()**



在文字資料發出前發出的信號。

以下是您可以選擇的語言和國家代碼，編碼是先以語言別區分，再來才列出使用該語言的區域。

- ces (Czech) 捷克語
  - CZE

- spa (Spanish) 西班牙語

- ESP
- USA

- deu (German) 德語

- AUT
- BEL
- CHE
- DEU
- LIE
- LUX

- fra (French) 法語

- BEL
- CAN
- CHE
- FRA
- LUX



B

- nld (Dutch) 荷蘭語

- BEL

- NLD

- ita (Italian) 義大利語

- CHE

- ITA

- pol (Polish) 波蘭語

- POL

- eng (English) 英語

- AUS

- BEL

- BWA

- BLZ

- CAN

- GBR

- o HKG
- o IRL
- o IND
- o JAM
- o MHL
- o MLT
- o NAM
- o NZL
- o PHL
- o PAK
- o SGP
- o TTO
- o USA
- o VIR
- o ZAF
- o ZWE



TinyWebDB

圖形

TinyWebDB



功能

非可視元件，藉由網路服務通訊來儲存與提取資訊。

請參考 <http://www.appinventorbeta.com/learn/reference/other/tinywebdb.html>

屬性

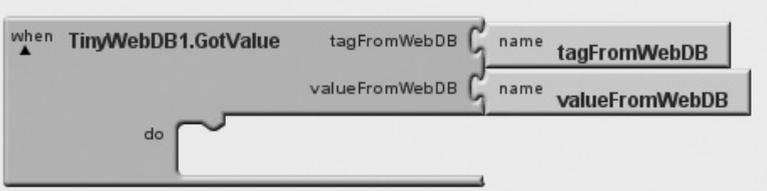
ServiceURL



事件

GetValue(text tagFromWebDB, any valueFromWebDB)

表示 GetValue 伺服器的請求已達成。



ValueStored()

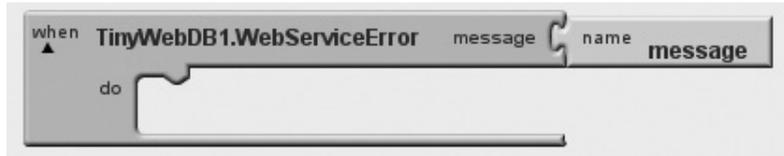
表示 StoreValue 伺服器請求的事件已達成。



TinyWeb-DB

**WebServiceError(text message)**

表示與網路伺服器通訊時信號發生錯誤。



方法

**GetValue(text tag)**

GetValue 要求網路伺服器取得指定標籤下的指定數值，如果該標籤下沒有儲存任何數值，則網路伺服器會自行決定要回傳什麼，此元件會接收任何回傳的資料。



**StoreValue(text tag, any valueToStore)**

要求網路伺服器儲存指定標籤下的指定數值。



