

# 使用 Zenbo 實驗室

## 什麼是 Zenbo 實驗室

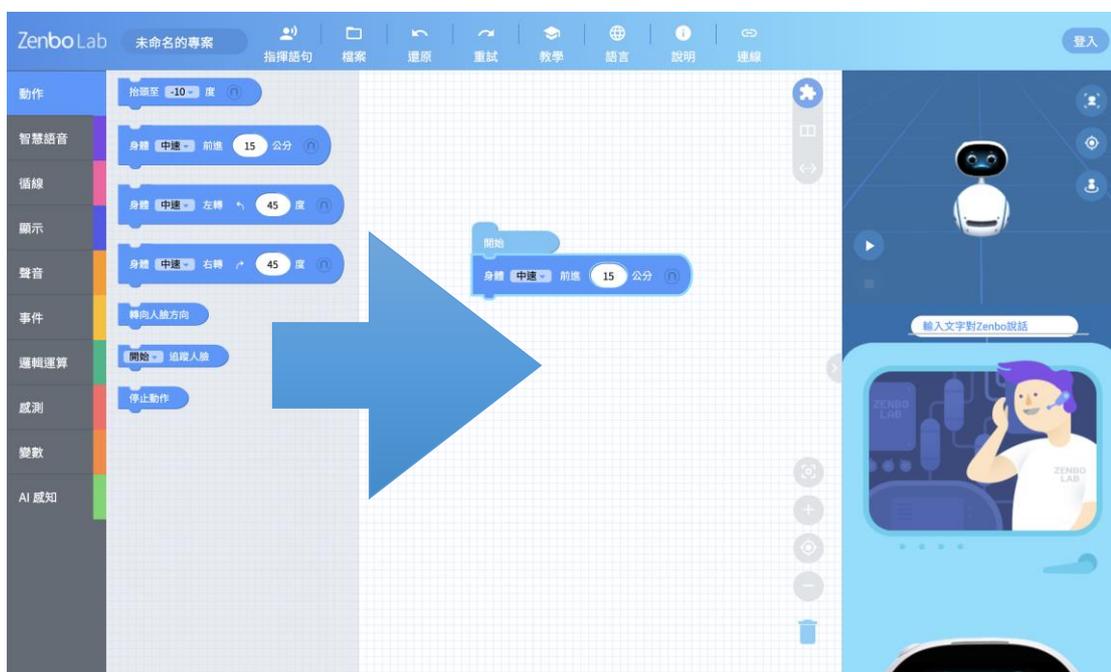
Zenbo 實驗室是一款圖形化的編程軟體，可以學習智慧語音、視覺辨識等等人工智慧編程，並且支援 Python 編譯。獨特的機器人模擬器系統，可以在畫面上模擬 Zenbo Junior 的行為。除了學習 STEM 課程，Zenbo 實驗室強大的能力更可以讓 Zenbo Junior 協助其他課程，像是英語對話、趣味問答等等。

Zenbo 實驗室的檔案可以單獨在 Zenbo Junior 上運行，把編輯好的檔案放進 Zenbo Junior 後隨時都可以執行，不再需要電腦。

利用智慧語音系統，可以用語音呼叫出不同的 Zenbo 實驗室檔案來執行。像是可以設定檔案的指揮句為“自我介紹”。想要執行的時候，只要喚醒 Zenbo Junior 並對他說“自我介紹”便可以執行自己設計的自我介紹內容喔！

## 1. 開始使用

簡單五步驟就可以控制 Zenbo Junior！讓 Zenbo Junior 往前走！



第一步：將左邊選單中的積木拖移到工作區域，組合好想要的動作與說話內容

第二步：打開 Zenbo Junior 上的 Zenbo 實驗室。

可以看到這台 Zenbo Junior 的 IP 與網路名稱

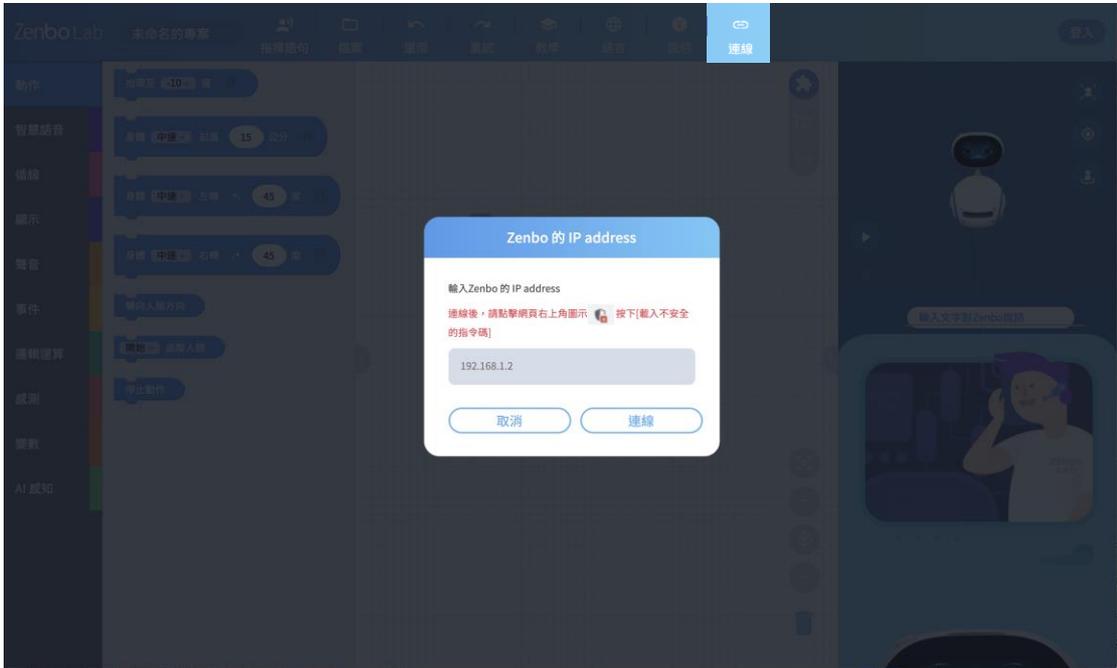


第三步：點選連線，輸入 Zenbo Junior 的 IP 並選擇連線！

第四步：點選 Chrome 右上方的盾牌圖示，再按載入不安全指令碼。



第五步：點選連線，輸入 Zenbo Junior 的 IP 並選擇連線！

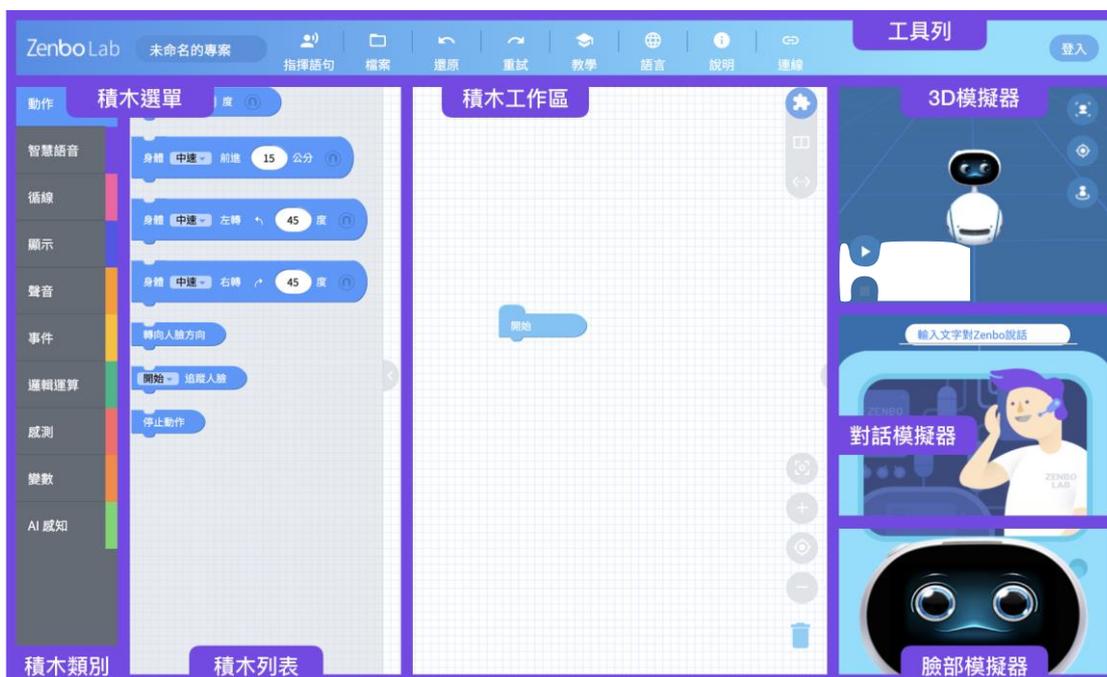


以上五個步驟做完後，再按連線按鈕就可以看到 Zenbo Junior 做出你想要的動作了！

## 2. 執行 Zenbo 實驗室的系統需求

Zenbo 實驗室不需安裝，在網頁上即可使用。建議使用 Chrome 瀏覽器。

### 3. 瞭解 Zenbo 實驗室的介面



#### 3.1. 工具列

##### 3.1.1. 專案名稱

預設的檔案名稱是「未命名的專案」，隨時可以在這裡修改。

##### 3.1.2. 設定指揮句

當想要對 Zenbo Junior 用語音呼叫執行特定專案時，可以在這裡設定想要的指揮句。例如設計了一個專案內容是 Zenbo Junior 跳芭蕾舞，並設定“跳芭蕾舞”為指揮句。把專案傳輸到 Zenbo Junior 以後，只要對 Zenbo 說“嗨，小布！跳芭蕾舞”，Zenbo Junior 就會開始跳芭蕾舞了！

每個專案可以設定 10 個指揮句子。

##### 3.1.3. 檔案

這裡可以新建專案、開啟舊檔案、存檔與另存新檔。標題的檔名顯示也可以直接修改檔名。匯出 Python 的功能可以匯出 Python 的 Py 檔案。

##### 3.1.4. 還原與重試

還原可回到編輯的上一步，重試可以重做剛剛的編輯。

##### 3.1.5. 教學

可連結到 Zenbo 實驗室功能說明網頁。  
<連結>

##### 3.1.6. 語言

切換介面顯示語言。Zenbo 實驗室支援繁體中文、簡體中文與英文。

### 3.1.7. 說明

你可以在此找到軟體版本與使用者授權同意書。

### 3.1.8. 連線

連線到 Zenbo Junior 執行專案，請輸入 Zenbo Junior 的 IP。你可以在 Zenbo Junior 的 Zenbo 實驗室畫面找到 IP 資訊，請確認編輯專案使用的電腦與 Zenbo Junior 在相同的無線網路下。

### 3.1.9. 登入

點選工具列最右邊來登入帳戶。

登入帳戶可以使用更多功能，例如智慧偵測的聲紋辨識與人臉辨識。除了註冊 Zenbo 實驗室專屬帳戶外，還可以使用 Facebook 與 Google 的帳戶來登入。

## 3.2. 積木選單

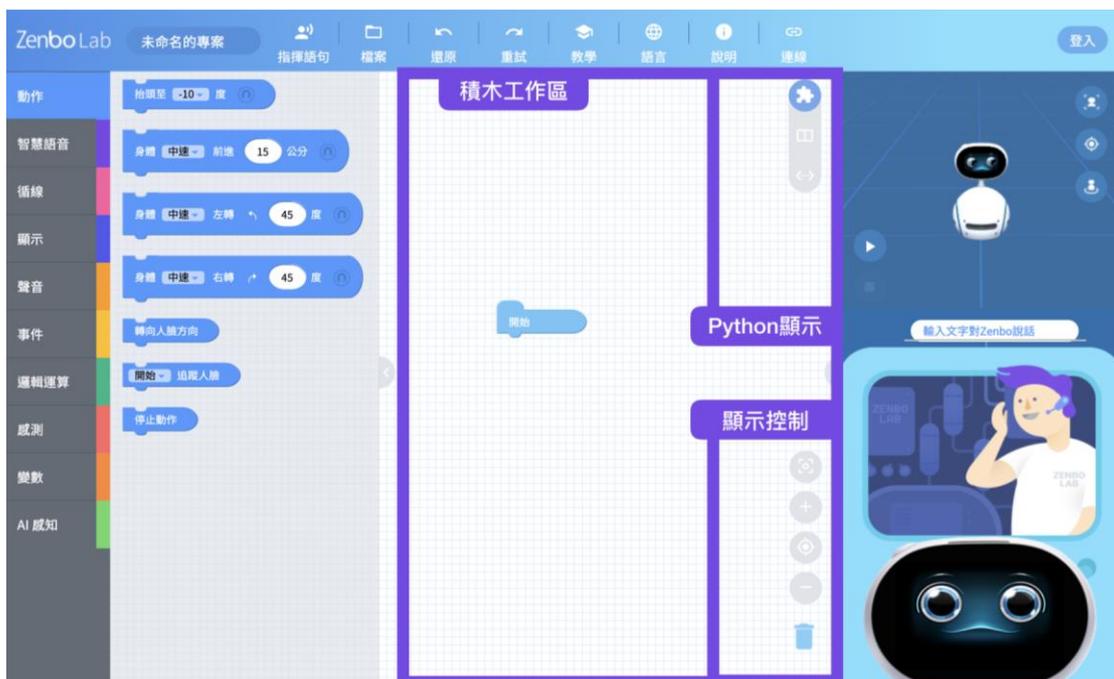
這裡可找到所有實驗室支援的程式積木功能。個別的積木內容在後續的章節有[詳細介紹](#)。

### 3.2.1. 積木類別

這裡列出所有積木類別，點選類別項目可以打開類別的積木列表。點選積木列表最右邊的箭頭可以把積木列表收起來。

### 3.2.2. 積木列表

列出這個類別的積木，如果積木很多，可以使用邊緣的拉把來移動。



## 3.3. 積木工作區域

這裡是真正進行編輯程式的地方，把左邊的積木選單與右邊的模擬區收起來可以獲得最大的工作區域。

### 3.3.1. 組合積木

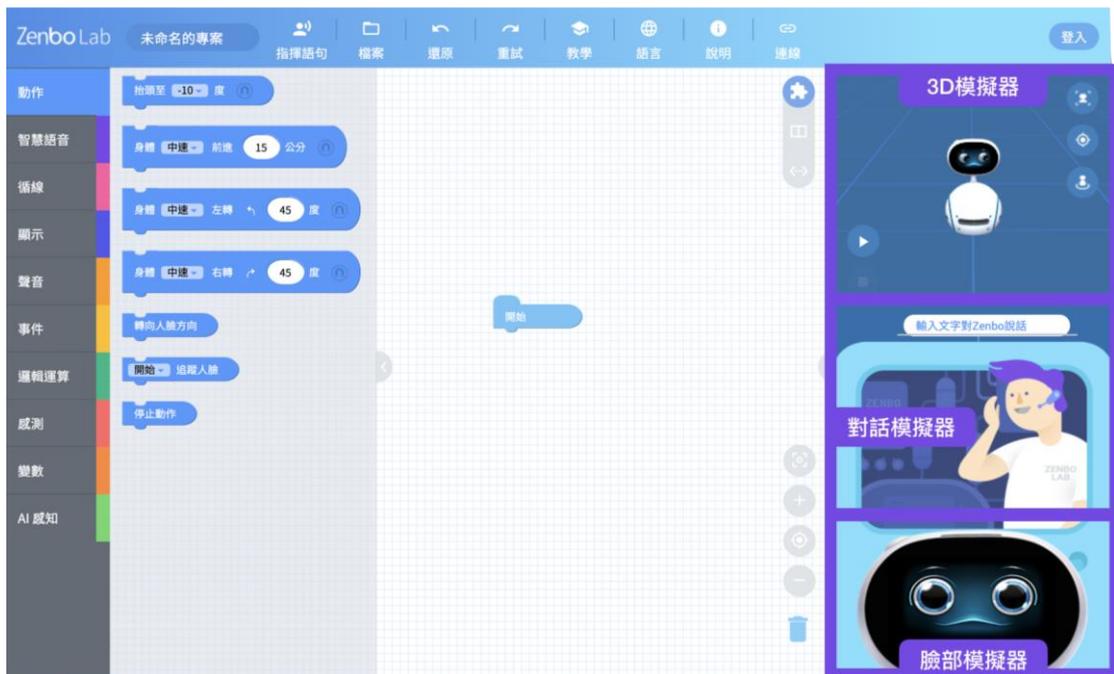
可以將積木選單中的積木拖拉到這裡進行組合，設計想要的行為與動作。

### 3.3.2. 刪除積木

要刪除積木可以把積木拖放到垃圾桶、拖到積木選單或是滑鼠右鍵選擇刪除，都可以刪除積木。

### 3.3.3. 工作區域顯示控制

除了可以使用滑鼠滾輪放大縮小積木，也可以使用控制按鈕，要是積木太多也可以利用回到中心點或是顯示所有積木的按鈕。



## 3.4. Zenbo 模擬器

### 3.4.1. 3D 模擬器

連接到 Zenbo Junior 之前，這裡可以模擬 Zenbo Junior 的動作與行為。點擊播放鍵即可開始模擬，點擊停止則停止模擬。三個輔助按鍵可以幫助重新定位模擬器中的 Zenbo、重置畫面或用頂視圖視角來觀看。

### 3.4.2. 對話模擬器

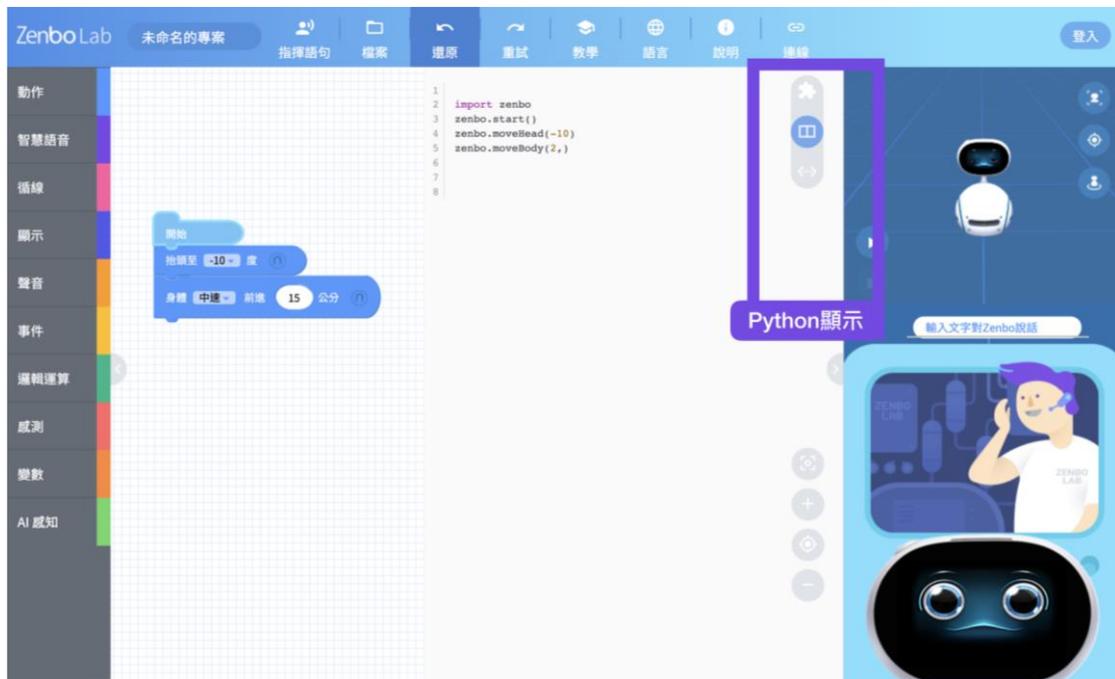
要模擬對 Zenbo Junior 說話時候，可以用鍵盤輸入文字在對話框裡來模擬聽到的句子。

### 3.4.3. 臉部模擬器

點選一下 Zenbo 頭頂可以開啟臉部顯示模擬器，再點一下可以收合。這裡可以模擬顯示 Zenbo 的表情與設計的介面。



## 4. Python



### 4.1. 查看專案的 Python 內容

點選 Python 觀看按鈕打開 Python 程式語言內容。

每一個積木都可以對應到 Python 的程式碼，要觀看所編輯的程式碼，可以利用積木-Python 程式對應模式，三種觀看模式：全積木、積木與 Python 碼對照、全 Python 碼，也可以將此處的程式複製到其他文字編譯器使用。

### 4.2. 使用 Python 語言控制 Zenbo

如果想要撰寫 Python 語言來控制 Zenbo Junior，您需要下載[官方編譯器](#)，下載完畢安裝後，使用官方的編譯環境來撰寫相關程式碼。

當您需要連線到 Zenbo Junior 時，請確認 Zenbo 機器人開啟了 Zenbo 實驗室 App，並且電腦與 Zenbo Junior 在同一個無線網路底下。

## 5. 積木形狀

### 5.1. 控制積木

前後都可以堆疊積木的基礎形狀，他們有可能是長條形的或是C夾型的。



### 5.2. 起始型事件積木

起始型積木只能往下堆疊一個新的程式，而如何開始執行這段程式，則看起始型積木內所寫的條件，許多不同的起始積木下的程式可以在不同的時機執行。



### 5.3. 邏輯判斷布林積木

積木裡面寫的條件都會得到一個成立或是不成立的結果，所得到的值就是 True 成立或是 False 不成立。這個積木必須要放到其他積木相對應形狀裡面才有作用。



### 5.4. 變數型積木

每個變數型機積木都會有一個值，可能是數字型的也可能是字串型的。數字型的變數表示裡面儲存的是純數值，沒有文字或是符號，可以拿來計算，字串型的變數則表示裡面儲存的是文字或是符號。這個積木必須要放到其他積木相對應形狀裡面才有作用。



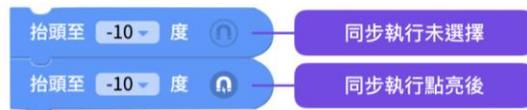
## 6. 基礎積木功能

### 6.1. 開始

每個專案的積木群都由開始積木開始，當專案被執行的時候，會從開始積木往下一個一個積木執行。

### 6.2. 同步執行與積木運行規則

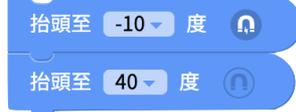
所有的積木都是執行完畢後才會執行下一個積木。但有些積木的最右方有同步執行的圖示，點亮圖示後，下一個積木就會跟著這個一起同步執行。



像是如果想要一邊往前走一邊說話，就可以點亮身體移動積木的同步執行來達成，如果沒有點亮，則 Zenbo Junior 會走完才說話。



如果上下兩個積木控制同樣的資源，像是都是控制頭部，那麼即使點亮了同步執行的選項，還是會先完成上面的積木才會進行下面的積木。



要強制停止執行積木，進行其他的積木，則需要使用到事件積木觸發。



當手碰觸螢幕往下滑的時候觸發了這個積木，這時候就會打斷抬頭 40 度的動作，而改執行向下-10 度的動作。

### 6.3. 動作

#### 6.3.1. 抬頭

可以控制 Zenbo Junior 頭的角度。

例：抬頭至(-10)度，Zenbo Junior 頭向下到-10 度的位置。

輸入 0 度時，機器人的頭會朝向正前方，頭部上下的角度限制是-10 度到 50 度之間。

#### 6.3.2. 身體移動

可以控制 Zenbo Junior 移動的速度以及距離。  
例：身體(中速)前進(15)公分，Zenbo 機器人向前移動 15 公分。  
輸入數值的上下限為正負 30,000。若輸入負值的話，機器人會轉向到背後再前進。

### 6.3.3. 身體向左轉

可以控制 Zenbo Junior 轉動的角度。  
例：身體(中速)左轉(45)公分，Zenbo 機器人向左轉 45 度。  
輸入數值的上下限為正負 3,000。若輸入負值的話，機器人會向右轉。

### 6.3.4. 身體向右轉

可以控制 Zenbo Junior 轉動的角度。  
例：身體(中速)右轉(45)公分，Zenbo 機器人向右轉 45 度。  
輸入數值的上下限為正負 3,000。若輸入負值的話，機器人會向左轉。

### 6.3.5. 轉向人臉的方向

如果偵測到人臉，會將 Zenbo Junior 的臉轉到面對人臉方向，但不會跟著人臉移動。

### 6.3.6. 追蹤人臉

Zenbo Junior 會開始小幅度的轉動尋找附近有沒有臉，如果偵測到了就會跟著人臉的位置轉動，但是 Zenbo Junior 不會離開原來的位罝。

### 6.3.7. 停止動作

可以停止 Zenbo Junior 身體與頭的動作。

## 6.4. 顯示

### 6.4.1. 做出表情

讓 Zenbo Junior 做出想要的表情，一共有 24 個表情可以選擇，切換了表情後，這個表情會持續呈現，直到再度切換別的表情。

### 6.4.2. 輪子 LED 的燈光顯示

設定雙輪或是單輪 LED 燈光顯示，顯示方式共有 14 種模式，其中靜態、呼吸、閃爍、彗星、移動-閃光、單色波浪六種模式可以選擇燈光的顏色。其他 8 種則是特殊變化無法選擇顏色。  
設定好燈光效果後就會持續一直呈現，直到改變顯示狀態或是關閉燈光。

### 6.4.3. 停止輪子 LED 的燈光效果

可以設定停止雙輪或是單輪的燈光效果

### 6.4.4. 錄影

需要設定錄影的長度，可以選擇 15 秒、30 秒、45 秒、60 秒的長度，這個功能只有在完成專案傳送到 Zenbo Junior，在 Zenbo Junior 上執行才有作用。

### 6.4.5. 停止錄影

強制停止錄影的功能，可在所設定錄影的秒數使用完畢前先停止錄影。但必須配合事件積木的積木一起使用，請見[同步執行](#)中積木運行規則的說明。

#### 6.4.6. 播放剛錄製的影片

設定開始播放或是停止播放剛剛錄製的影片。停止功能必須配合事件積木的積木一起使用，請見[同步執行](#)中積木運行規則的說明。

#### 6.4.7. 照張相片

使用照相積木，會開啟相機，倒數三秒後拍照。這個功能只有在完成專案傳送到 Zenbo Junior，在 Zenbo Junior 上執行才有作用。

#### 6.4.8. 顯示剛拍的照片

設定開始播放或是停止顯示剛拍的照片。

#### 6.4.9. 播放圖片或影片

點選檔案來源來指定想要播放的圖片或是影片檔案位置。停止功能必須配合事件積木的積木一起使用，請見[同步執行](#)中積木運行規則的說明。

### 6.5. 聲音

#### 6.5.1. 錄音

需要設定錄音的長度，可以選擇 30 秒~180 秒的長度。

#### 6.5.2. 停止錄音

強制停止錄音的功能，可在所設定錄音的秒數使用完畢前先停止錄音。但必須配合事件積木的積木一起使用，請見[同步執行](#)中積木運行規則的說明。

#### 6.5.3. 播放剛錄製的聲音

設定開始播放或是停止播放剛剛錄製的聲音，若沒有設定停止的時機，則播放至音檔結束。停止功能必須配合事件積木的積木一起使用，請見[同步執行](#)中積木運行規則的說明。

#### 6.5.4. 調整媒體音量

調整媒體音量比原本的大一級或是小一級。

#### 6.5.5. 設定媒體音量

設定媒體音量 0~100%。

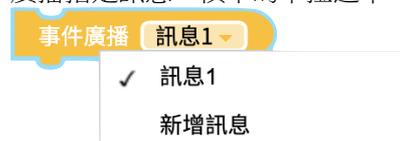
#### 6.5.6. 播放音樂

點選檔案來源來指定想要播放的音樂。停止功能必須配合事件積木的積木一起使用，請見[同步執行](#)中積木運行規則的說明。

### 6.6. 事件

#### 6.6.1. 事件廣播

廣播指定訊息，積木的下拉選單可以新增訊息。



### 6.6.2. 當收到訊息

收到廣播的指定訊息時，開始執行這個積木以下的項目。



如果收到廣播訊息肚子餓了，說出我肚子餓了。

### 6.6.3. 當摸頭時

在任何時候 Zenbo Junior 被摸頭時，開始執行這個積木以下的項目。

### 6.6.4. 當接上電源時

在任何時候 Zenbo Junior 接上電源線時，開始執行這個積木以下的項目。

### 6.6.5. 當拔除電源時

在任何時候 Zenbo Junior 拔除電源線時，開始執行這個積木以下的項目。

### 6.6.6. 當電池電量高於或低於某個百分比

在任何時候 Zenbo Junior 電池電量高於或低於某個百分比時，開始執行這個積木以下的項目。

### 6.6.7. 當臉部螢幕被觸碰時

在任何時候 Zenbo Junior 臉部螢幕被觸碰時，開始執行這個積木以下的項目。  
螢幕觸碰可以指定被觸及到的數量，從 0 點到 10 點同時被處碰。

### 6.6.8. 當手滑過螢幕時

在任何時候使用者在 Zenbo Junior 臉部螢幕滑動時，開始執行這個積木以下的項目。  
滑螢幕可以指定方向，向上、向下、向左、向右。

### 6.6.9. 當按下頭頂按鍵時

在任何時候頭頂上的按鍵被按下時，開始執行這個積木以下的項目。

### 6.6.10. 停止全部程式

中止執行專案，並且停止所有正在進行的動作、多媒體與語音。

## 6.7. 邏輯運算

### 6.7.1. 等待

等待指定的秒數後，開始執行這個積木以下的項目。

### 6.7.2. 重複

重複執行包含積木指定的次數。



重複說出我愛吃蘋果 5 次。

### 6.7.3.重複無限次

重複執行無限次包含的積木。



一直不停說出我愛吃蘋果。

### 6.7.4.如果，那麼

如果達成了指定條件，就執行包含的積木



如果 Zenbo Junior 正在充電中，說出我正在充電。

### 6.7.5.如果，那麼，否則

如果達成了指定條件，就執行包含的積木，否則就執行另外的積木。



如果 Zenbo Junior 正在充電中，說出我正在充電，否則說出請幫我充電。

### 6.7.6.重複直到

重複一直執行包含的積木，直到條件被滿足。

### 6.7.7.等待直到

等待到條件被滿足，開始執行這個積木以下的項目。

### 6.7.8.離開迴圈

強制離開執行中的重複迴圈。

### 6.7.9.相等

條件 A 等於 B 是不是成立。

### 6.7.10.大於

條件 A 大於 B 是不是成立。

### 6.7.11.小於

條件 A 小於 B 是不是成立。

### 6.7.12.相加

A 加 B 的結果。

### 6.7.13.相減

A 減 B 的結果。

### 6.7.14.相乘

A 乘 B 的結果。

### 6.7.15.相除

A 除 B 的結果。

### 6.7.16.且，兩個條件都成立

兩個條件都成立時，整體條件成立。



如果頭頂按鍵是被按下的，且同時螢幕有被一根手指碰觸到，說出我喜歡你。這兩個條件只會在執行到“如果”這個積木時被檢查。

### 6.7.17.或，其中一個條件成立

其中一個條件成立時，整體條件就成立。



如果頭頂按鍵是被按下的，或是螢幕有被一根手指碰觸到，任一個條件被滿足就說出我喜歡你。這兩個條件只會在執行到“如果”這個積木時被檢查。

### 6.7.18.條件不成立

指定條件不成立時，結果為真。



如果頭頂按鍵沒被按，會說出妳沒有按我的頭。

### 6.7.19.A 除以 B 的餘數

取出 A 除以 B 的餘數來使用。

### 6.7.20.隨機取數

在指定的區域間隨機取一個數值。

### 6.7.21.字串組合

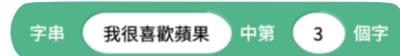
把兩個字串組合成一個字串。



新字串結果為香蕉很好吃。

### 6.7.22.字串的第幾個字

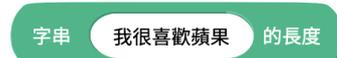
字串的第幾個字是什麼。



我很喜歡蘋果的第三個字是喜。

### 6.7.23.字串的長度

字串有幾個字。



我很喜歡蘋果有 6 個字。

### 6.7.24.字串是否包含某些詞



我很喜歡蘋果包含蘋果？ 這個條件成立。

### 6.7.25.其他的數學運算

取出數學運算的結果來使用。

數學運算包含：四捨五入值、絕對值、無條件捨去、無條件進位、平方根、  
sin、cos、tan、asin、acos、atan、ln、log。

## 6.8. 感測

### 6.8.1.正在充電中？

Zenbo 機器人目前是不是正在充電？是的話邏輯判斷結果為條件成立。



如果 Zenbo Junior 正在充電，說出我正在充電。

### 6.8.2.電量百分比

Zenbo Junior 目前的電量百分比是多少。



如果 Zenbo Junior 電量少於 50，說出我肚子有點餓，否則說出我吃飽了。

### 6.8.3. 媒體音量

多媒體的播放音量數值，範圍是 0~100。



說出目前的多媒體音量大小百分比。

### 6.8.4. 頭部被摸？

Zenbo Junior 的頭偵測時有沒有被摸？是的話邏輯判斷結果為條件成立。



如果被摸摸頭，說出我的頭有點癢。

### 6.8.5. 聲納偵測到某個方向有無障礙物？

可以指定聲納偵測的方向，中間、左邊、右邊或是任何方向有沒有障礙物？是的話邏輯判斷結果為條件成立。



運行到“如果那麼”這個積木時，如果任何方向有障礙物，向後轉走 50cm。

### 6.8.6. 聲納偵測到某個距離有無障礙物？

可以指定聲納偵測的距離有沒有障礙物？是的話邏輯判斷結果為條件成立。可以指定 30~60 公分間每五公分的距離。



運行到“如果那麼”這個積木時，如果 30 公分前有障礙物，向後轉走 50cm。

### 6.8.7.聲音來自什麼方向

運行到這個積木時，藍耳朵會打開一秒鐘，聽音辨位偵測這時候的聲音方向，可以指定辨識是不是 Zenbo Junior 的前方、右方、後方、左方。



如果聲音來源右方，說出敵人在右邊！。

### 6.8.8.聲音來源的角度

運行到這個積木時，藍耳朵會打開一秒鐘，聽音辨位偵測這時候的聲音方向，正前方是 0 度、正右方是 90 度、正後方是 180 度、正左方是 270 度。角度的範圍由 0 度到 359 度。



把身體轉向聲音的角度。

### 6.8.9.頭頂按鍵被按下？

運行到這個積木時，頭頂的按鍵有沒有被按下？是的話邏輯判斷結果為條件成立。

### 6.8.10.音量鍵+被按下？

運行到這個積木時，音量增加鍵有沒有被按下？是的話邏輯判斷結果為條件成立。

### 6.8.11.音量鍵-被按下？

運行到這個積木時，音量減少鍵有沒有被按下？是的話邏輯判斷結果為條件成立。

### 6.8.12.手指觸碰螢幕？

運行到這個積木時，臉有沒有被碰觸？是的話邏輯判斷結果為條件成立。必須指定臉有多少點被碰觸到，由 1 點到 10 點。



如果臉被碰觸到一點，則說出有一根手指碰我的臉。此時如果用兩根手指碰觸螢幕，也一樣不會說出。

## 6.9. 變數

### 6.9.1. 設定數字變數

數字型的變數表示裡面儲存的是純數值，沒有文字或是符號，可以拿來計算。設定數字變數的數值與名稱。數值可以是負數或是小數。



新增數字變數名稱我的數學成績為 100。

### 6.9.2. 設定字串變數

字串型的變數表示裡面儲存的是文字或是符號。設定字串變數的內容與名稱。可以輸入中文、英文、數字以及空白鍵、逗號、句號、驚嘆號、問號但是不能輸入其他特殊符號。



設 同學的外號 為 神奇超人

新增字串變數名稱同學的外號為神奇超人。

### 6.9.3. 數字變數

數字型的變數表示裡面儲存的是純數值，沒有文字或是符號，可以拿來放在其他積木做運算或使用。

### 6.9.4. 字串變數

字串型的變數表示裡面儲存的是文字或是符號，可以拿來放在其他積木做使用。

### 6.9.5. 數字變數改變量

設定數字變數的增加或是減少，如果填入正數就會增加，負數則減少。

## 7. 智慧語音積木

### 7.1.1. 說出語句

說出寫入的語句



說出我喜歡你

### 7.1.2. 調整說話速度

有三個速度可以調整，緩慢、正常跟飛快

### 7.1.3. 改變說話聲調

聲音可以變化為低沈、正常或是高亢

### 7.1.4. 用中文口音或英文口音說話

用中文口音講英文的時候，聽起來可能會較不順暢，但是可以在同一個句子裡包含中文跟英文。如果使用英文口音講話，只能講英文的語句，若是中英文混合的句子，中文會被略過。

### 7.1.5. 打開耳朵開始聽

要跟 Zenbo Junior 對話的時候，必須幫他開啟耳朵，告訴他何時開始聽。

### 7.1.6. 當聽到指定的語句

在任何時候當 Zenbo Junior 的耳朵打開，機器人聽到指定的語句時，開始執行這個積木以下的項目。



任何時候，當 Zenbo Junior 的耳朵打開，聽到誰是世界上最美麗的人，回答是你喔！

### 7.1.7.我剛剛聽到的語句

紀錄下來 Zenbo Junior 所聽到的語句。



Zenbo Junior 詢問你喜歡吃什麼水果呢？聽到回答後比對話裡面有沒有包含蘋果兩個字。如果包含蘋果，Zenbo Junior 回應我也最喜歡吃蘋果了。

### 7.1.8.關閉喚醒詞

藍耳朵可以用喚醒詞“Hey Zenbo!”或是“嗨 小布！”來打開，讓 Zenbo Junior 開始聽，但是有時候會希望 Zenbo Junior 不要被外界干擾或被其他人使用喚醒詞打開耳朵，這時候可以使用這個積木來關閉喚醒詞功能。

### 7.1.9.調整說話音量

調整說話音量比原本的大一級或是小一級。

### 7.1.10.設定說話音量

設定說話音量 0~100%。

## 8. 循線積木

### 8.1.1.開始循線

打開循線功能，這個積木執行後，將 Zenbo Junior 放在循線圖樣的平面上就會開始循著黑線行走不停止。

### 8.1.2.設定循線規則：偵測到單一顏色時的行為

設定偵測到單一顏色時，更新行為規則。



循線時前進時，偵測到紅色線以後，下次遇到叉路時都選擇往左。

### 8.1.3.設定循線規則：偵測到顏色組合時的行為

設定偵測到顏色組合時，更新行為規則。

顏色組合必須由系統色黃色開始，後面可使用一到三個顏色。  
例：黃紅藍綠或是黃紅。



循線時前進時，偵測到黃紅藍綠的顏色組合，下次遇到叉路時都選擇往右。

#### 8.1.4. 停止時，當循線感測器偵測到單一顏色

判別循線感測器偵測到的顏色。



偵測到紅色的時候，說出偵測到紅色了，並且一直執行這個偵測。

#### 8.1.5. 當循線感測器偵測到單一顏色

在任何時候偵測到紅、藍或綠其中一個顏色時，開始執行這個積木以下的項目。通常使用在循線動作以外的功能，像是一邊循線一邊說話或是轉頭。



循線時前進時，偵測到紅色，說出我闖紅燈了。

#### 8.1.6. 當循線感測器偵測到顏色組合

在任何時候偵測到顏色組合時，開始執行這個積木以下的項目。通常使用在循線動作以外的功能，像是一邊循線一邊說話或是轉頭。

顏色組合必須由系統色黃色開始，後面可使用一到三個顏色。  
例：黃紅藍綠或是黃紅。



循線時前進時，偵測到黃紅藍綠的顏色組合，說出我超速了。

#### 8.1.7. 停止循線

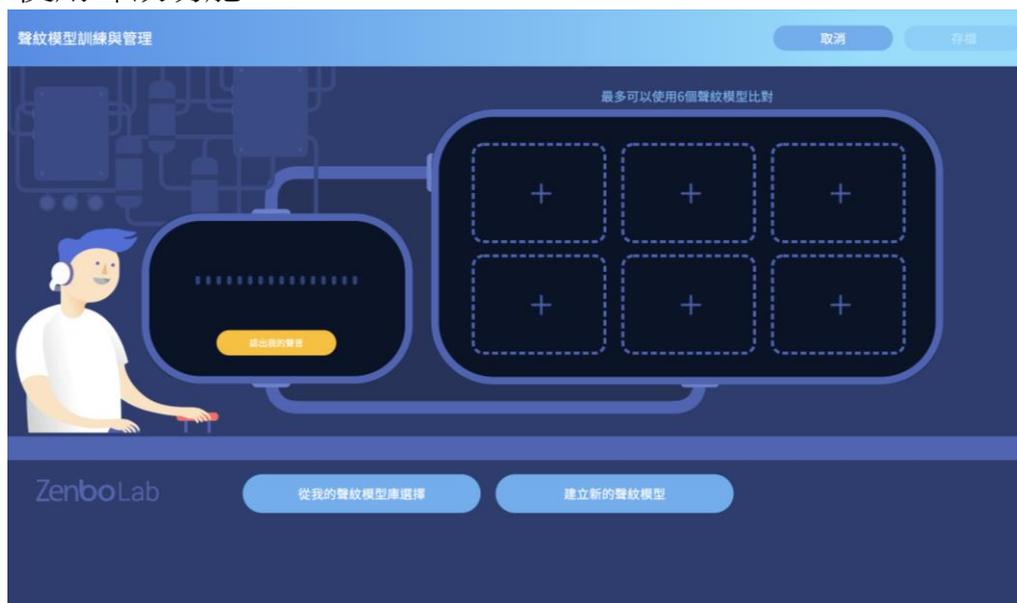
執行到這個積木時，才會停止循線的功能。

## 9. AI 感知積木

### 9.1. 聲紋辨識

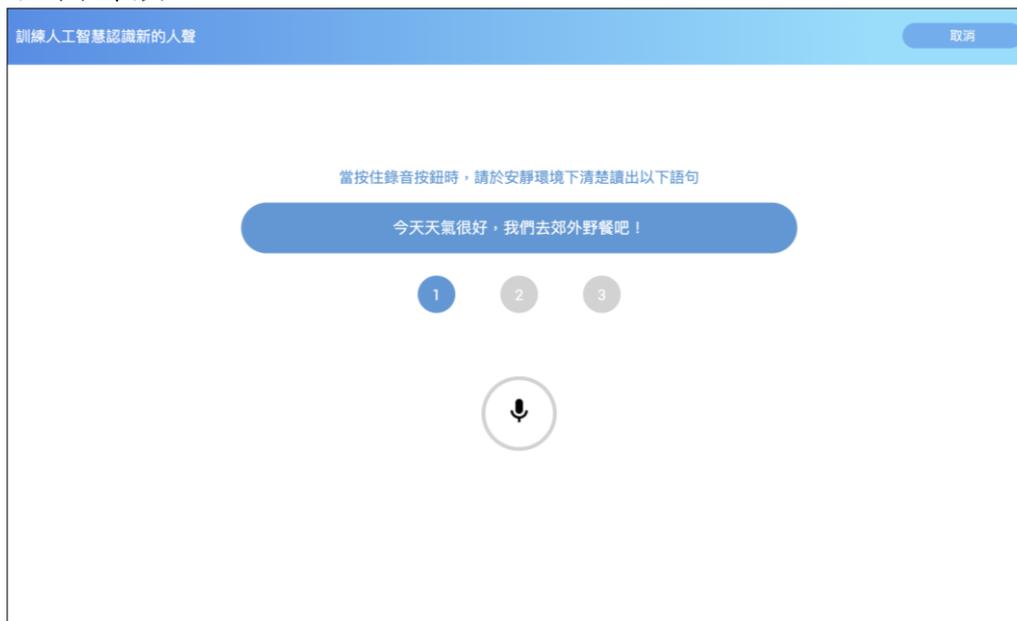
聲紋辨識的功能必須登入帳戶後才能使用，首次使用的時候會需要提供麥克風的權限。若要辨識聲紋，必須先設計開啟藍耳朵聽取聲音的情境，聲紋辨識相關積木都是分析上一次耳朵打開時聽到的聲音。

#### 9.1.1. 使用聲紋功能



點選註冊與管理你想要辨識的人聲後開啟管理畫面，每個專案最多可辨識六個人的聲音。註冊好聲紋以後，可以在管理畫面裡做測試，對麥克風講話就會辨識出聲紋的結果。

#### 9.1.2. 註冊聲紋



點擊畫加號或是建立新的聲紋模型按鈕來註冊聲紋，註冊聲紋需要每位註冊者對麥克風說出指定的三句話，依照圖示流程，輸入聲紋註冊者的暱稱，暱稱的

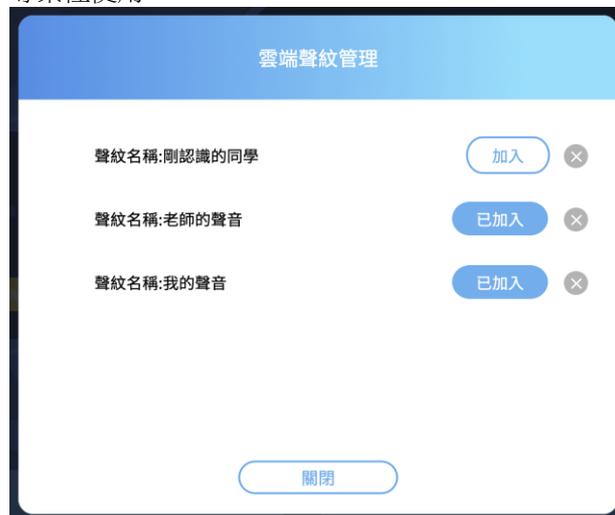
名稱需要與專案中使用的與曾經註冊過的其他聲紋暱稱不同，說出語句時需要一直按壓著錄音鍵，依照指示說出句子後放開錄音鍵，最後給這個聲紋取一個暱稱。



### 9.1.3. 從我的聲紋模型庫選擇

每次註冊完新的聲紋，會自動將聲紋資料放到曾經註冊過的檔案庫裡，每組帳戶最多可以記錄 10 筆註冊的聲紋資料，一但超過上限，會需要手動刪除舊有的資料才能再註冊新的聲紋資料。

當需要取用曾經註冊過的聲紋時，點選圖示，選取使用，即可將舊有聲紋在新專案裡使用。



### 9.1.4. 刪除聲紋

刪除專案裡的聲紋並不影響註冊過的檔案庫的資料，點選刪除圖示即可刪除。

### 9.1.5. 剛剛誰在說話

辨識說話的人是誰，這個變數積木會儲存一個人名暱稱。



小明對 Zenbo Junior 說猜猜我是誰？Zenbo Junior 辨識到實際說話的人是小明，回應說，我知道你是小明！

### 9.1.6. 剛剛說話的是誰的可能性 %

聲紋比對是將收錄進來的聲音跟專案裡指定的最多六組聲紋做比對，因此收錄的聲音可能會有下面的情況。90%跟小明的聲音相似，但小明跟小華的聲音很像，所以也有 80%跟小華相似。因此可能性並非所有人加起來為 100%。



Zenbo Junior 打招呼說你今天好嗎？小明或是小華回應我很好。如果小明跟小華的可能性相減的絕對值小於 10，Zenbo Junior 回問，請問你是小明還是小華呢？

### 9.1.7. 剛剛說話是指定的人嗎

這個積木會得到條件成立或是不成立的結果。



說話的人是小明的話，跟小明說我有個秘密要告訴你，否則說我的秘密不能告訴任何人。

## 10. 錯誤訊息與排除

### 10.1. 在 Zenbo Junior 運行時

#### 10.1.1. 可能會摔落

Zenbo Junior 遇到了高度落差，他可能太靠近桌子邊緣或是在階梯旁邊。這時候因為安全原因，Zenbo Junior 會停止運行實驗室專案，請將 Zenbo Junior 移動到安全地點來使用。

### 10.1.2. 資料線未移除

Zenbo Junior 還連接著 USB 線，這時候所有的移動與轉動動作都無法執行，若要使用請將連接線移除。

### 10.1.3. 充電中

Zenbo Junior 還連接著電源線，這時候身體的移動與轉動動作都無法執行，若要使用請將連接線移除。

## 10.2. 在 Zenbo 實驗室的網頁編輯器中

### 10.2.1. 開啟不同帳戶的專案

如果專案包含特殊智慧偵測積木，像是聲紋辨識，若需要編輯專案內容，就需要專案所有者才能夠編輯，但如果只是想查看專案內容，即使專案不屬於自己也可以開啟觀看，此時會進入唯讀模式，所有積木與設定都無法變更。

### 10.2.2. 即將登出

如果在不同的電腦登入 Zenbo 實驗室，舊的登入會自動登出。

### 10.2.3. 請登入帳戶以使用智慧偵測功能

特殊智慧偵測積木像是聲紋辨識需要登入帳戶才能觀看或是使用，請登入帳戶或是創立新帳戶來使用。

### 10.2.4. 數值超過極限

某些積木的輸入數值是有上下限的，如果超過了，畫面會有警示，請依畫面調整輸入的數值。