

第一課：什麼是編程

編程存在於所有「智能」的東西之中，其實所謂編程都是在重覆一個過程，無論程式有多精密其實都是在體現Input和Output的關係。我們以向Input輸入一些東西，當到某些條件後，Output便執行動作。



舉一個最簡單的例子，我們是如何開電視的呢：

- Input（輸入）是遙控器上面的按鈕，Output（輸出）就是電視的螢幕
- Condition就是執行下一步的「條件」，例如如果沒有下雨的話我就會出門，「沒有下雨」就是Condition。在開電視裡，條件就是我們「按下」這一個動作

編程是什麼樣子的 - 由深入淺的編程路

Javascript

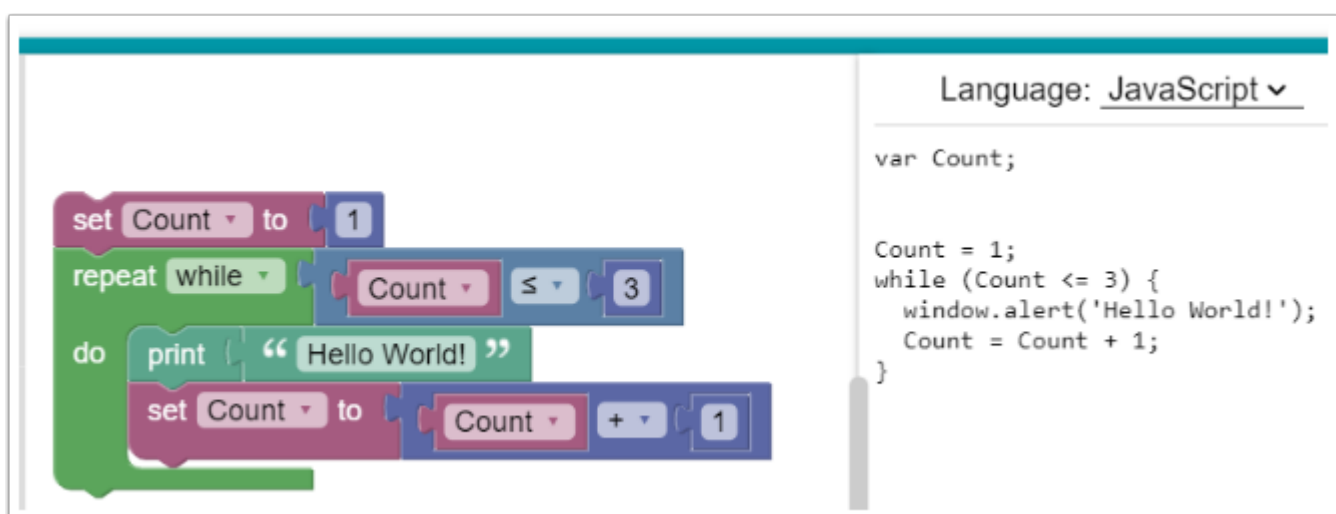
我們會由最深奧的編程開始，一步一步講到最簡單的模式，讓大家體驗由完全不明白到慢慢開始了解的過程。



這是Javascript，你正在上的網站很多也是透過它來構建的。你點擊某處會跳出什麼，你拉下拉動網站會如何scroll，全都是Javascript的語言在執行。在中學程度來說，學生最後可以自己寫出簡易的Javascript的，但還是需要數年以上的編程基礎。

對於完全未接觸過編程的人來說，這一大堆的編碼實在太困難，我也不打算詳細為大家一字字地拆解。為了解決這個困難，我們有了Blockly的誕生。

Blockly與micro:bit



大家自己對比一下左右，你會大概地知道其實兩者想做的是一模一樣的，只是左邊的外形看上去像lego一樣一格一格砌出來，我們稱之為Blocky形式的coding。

Blockly的好處在於我們可以稱虛幻的編程language轉化成易明的圖式，即係你未掌握如何編寫，你也可以大概地明白整條coding的基本意思。

micro:bit便正正是專門讓學生學習Blockly的工具，相信老師都或多或少聽過這個名字，因為它是現今STEM科裡最熱門的教材之一。

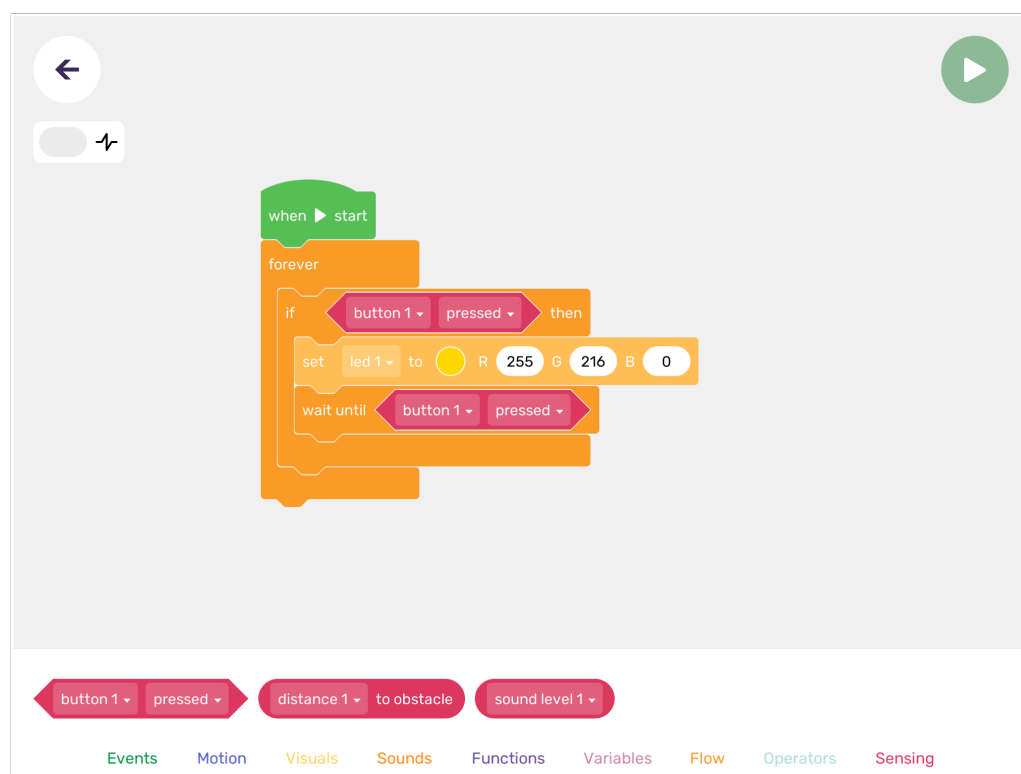
這是我們micro:bit code成品的例子（在我們專為產品提供的教學網站中便是這個樣子）。現在你可能看不明白當中發生了什麼事，但你也可能看得懂一部份，例如while和if then else這些字眼，你可以得知我們需要先達成某些條件才會執行下個動作。

說到這裡，還記得本文一開頭所講的嗎？「我們以向Input輸入一些東西，當到某些條件後，Output便執行動作。」便正正是編程的核心。

除此之外，你可以按動一下介面上方的轉換，可以快轉地Blockly語言至Javascript，這便是micro:bit最佳之處。

然而，說了這麼多，我們一開始也講明了本文並不是單純「micro:bit教學」。相信不是每個人也可以看懂上面micro:bit的內容吧，因此我們的旅程還沒有結束，讓我們繼續向更簡單的方向出發。

Robo Wunderkind



上面是Robo Wunderkind的Blockly App，是不是和micro:bit很像呢？卻又比micro:bit寫得更加淺白了，這次你可以直接根據上面的字面意思來了解到整個編程的目的。

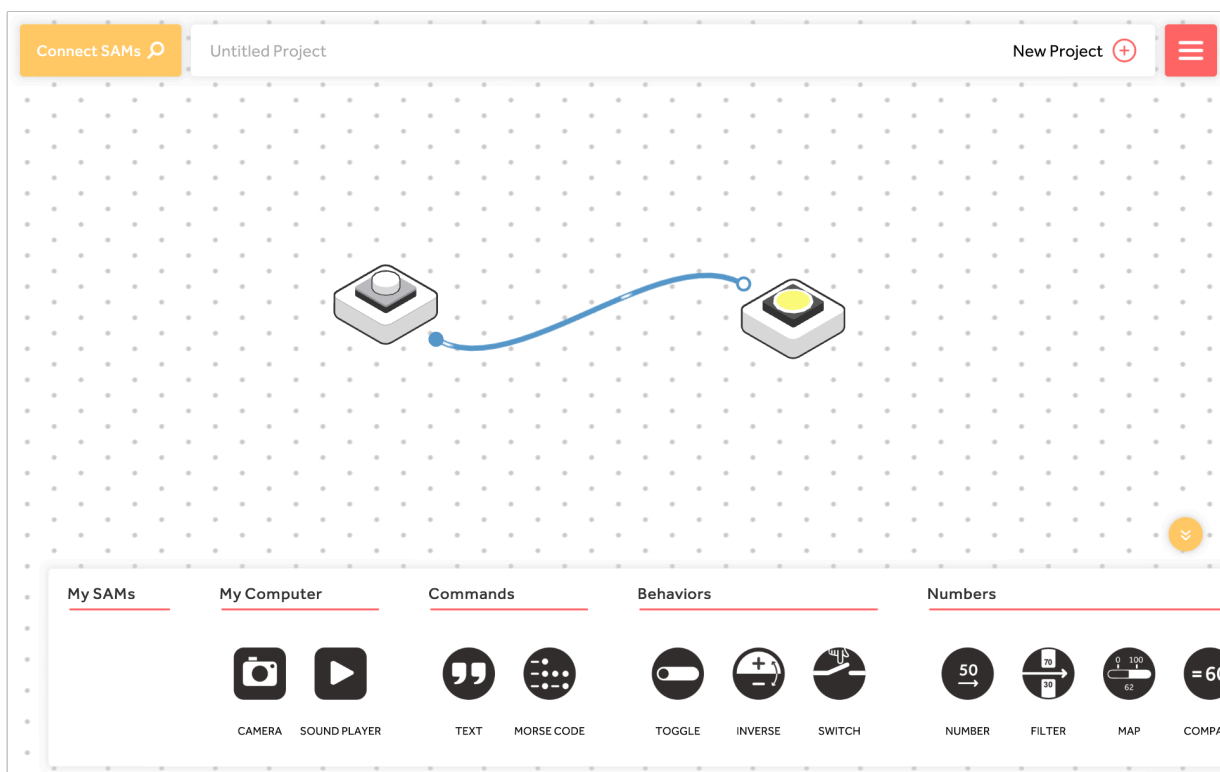
如果按鈕1被按下了，把燈轉成黃色，直到下一次按鈕1再次被按下。

但是，你可能到了這裡就會疑惑，為什麼我需要一個和micro:bit所此相近的教材？為什麼我不直接使用micro:bit？原因很簡單，因為這並不是Robo Wunderkind的全部。

SAM Labs與input、output



Robo Wunderkind對你來說可能已經足夠簡單了，但你或者會忽略了整篇文章一直圍繞着的基礎。還記得我們在一開始就說明了**output**和**input**的重要嗎？編程的語法固然重要，但SAM Labs就是讓學生掌握在這之前更加基礎的概念。



它是SAM Space App，它清楚地把上面不同的白色藍芽方塊分成input和output，圖中的例子就是最典型的input按鈕連接上output燈。

它的整個系列有著大部份你想像得到的input，如heat sensor、light sensor等，也可以使用這些input取得的數據來影響output的行動。在學習更複雜的sensor之前，先讓同學了解到什麼是collected data便是SAM Labs的最重要作用，否則他們如何能抽象中想出整個編程呢？



SAM Space我們放在文章的最尾段是故意的，因為它的容錯率十分高。為什麼這樣說？你看看上圖，其實這個方式的編程的起點（按鈕）和終點（燈）非常清晰，學生只需要在中間發揮創意，便可以一直嘗試至成功。這比起Blockly的語法，更加適合新手。

到了這裡，其實本文的所有教材我們將會在這10課中分別講解，而你應該也可以發現本文是特別設計過的。如果你由這裡反方向地閱讀至開頭，就是我們建議一個編程初學者的小朋友應該學習編程的次序。