開始使用 SAM Script

使用新程式碼平台的新手資源

Sam Labs總代理 廣天國際有限公司編印

目錄

在投影片模式，請點擊標籤以瀏覽各章節。

Sam Labs總代理 廣天國際有限公司編印

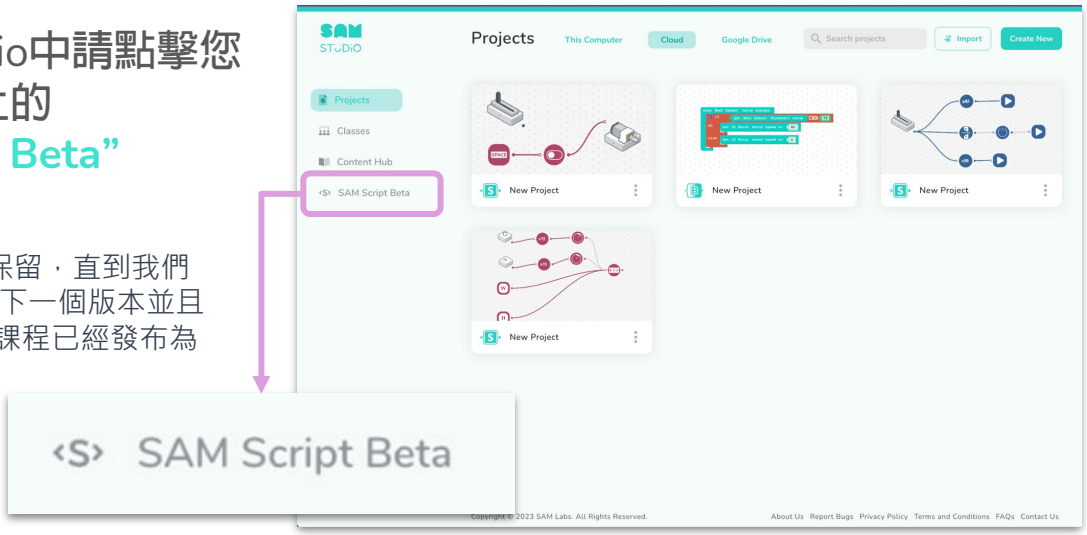
開始使用 SAM Script

我們很興奮地介紹一系列教師資源，旨在協助您在 SAM Studio 中更具吸引力且有效地開始使用 SAM Script。這些材料經過精心製作，旨在賦予您—我們的老師—所需的工具和知識，以無縫整合這個創新的編程平台到您課堂中。

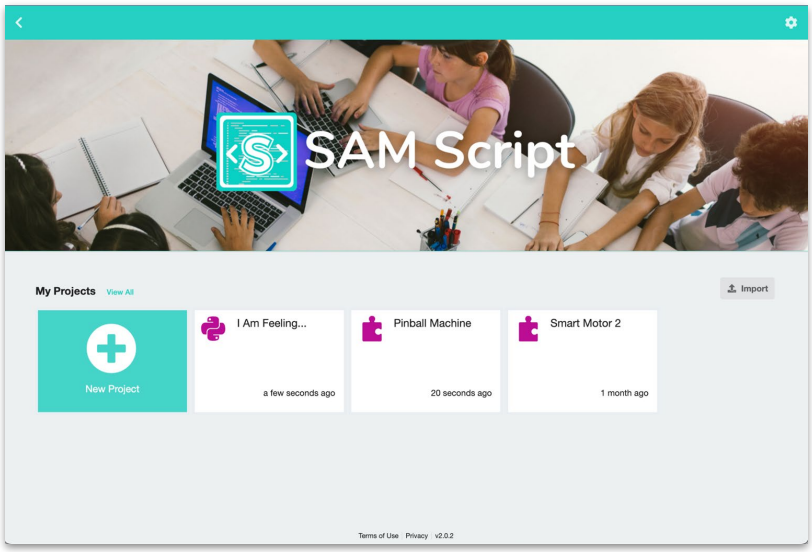
在 SAM Script建立新專案

1. 在 Sam Studio中請點擊您左側導覽列上的 **“SAM Script Beta”**

*「Beta」標籤將保留，直到我們更新 SAM Script 的下一個版本並且完整的6-8 STEAM 課程已經發布為止。



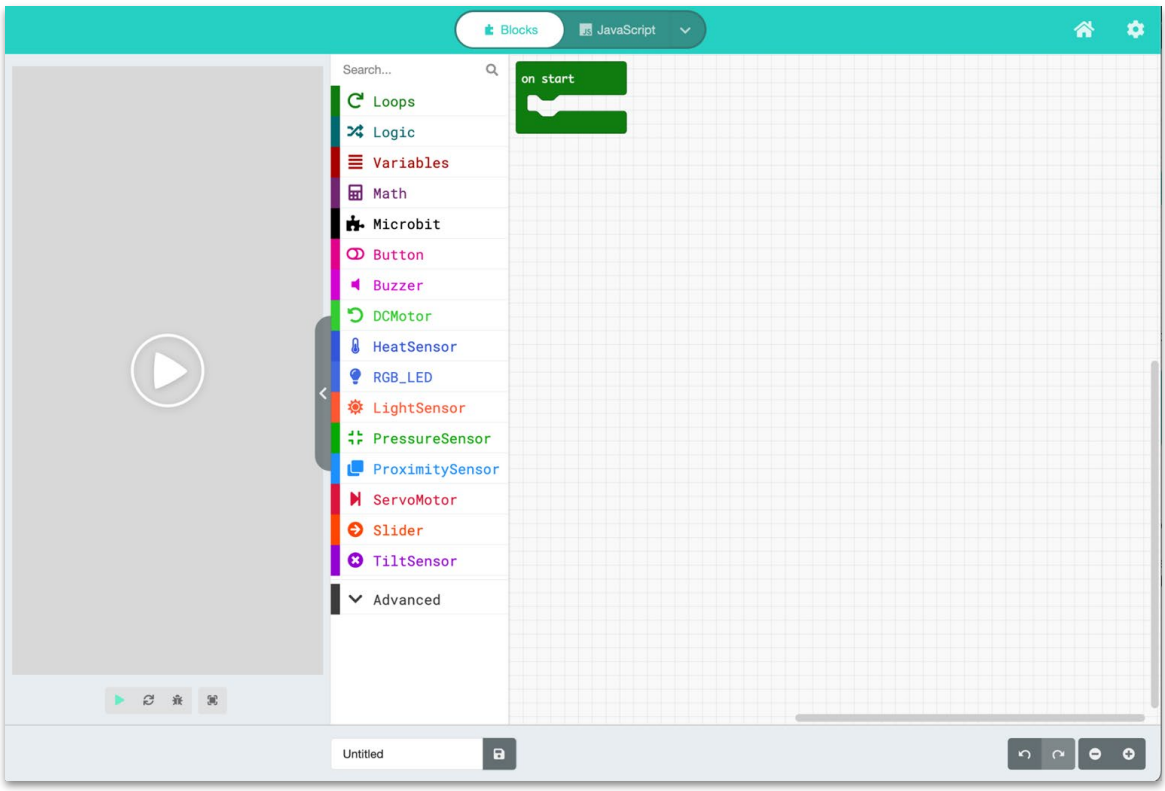
2. 這將在新視窗中開啟 SAM Script。畫面應該是這樣的：



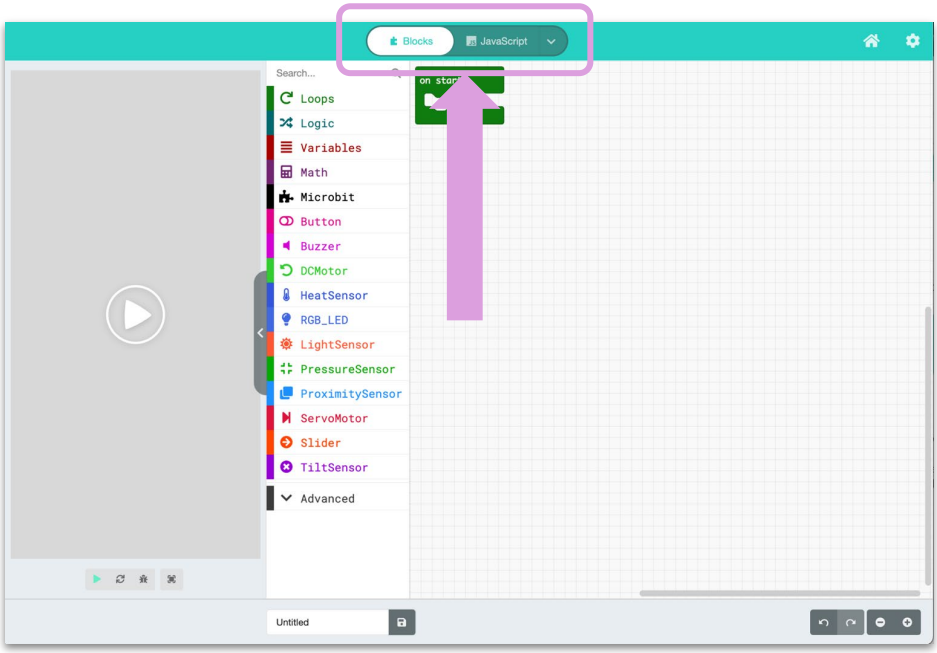
Sam Labs總代理 廣天國際有限公司編印

3. 在畫面左側，請點擊 “New Project”(新專案)按鈕。

4. 這將在新視窗中開啟 SAM Script。一個空白專案將被打開，畫面應該是這樣的：



5. 在畫面頂部中央選單中選擇編程語言：
   - Blocks
   - JavaScript
   - Python



這裡是選單的放大圖片。只需透過在選項之間切換，輕鬆點擊 Block-language 和 Script language

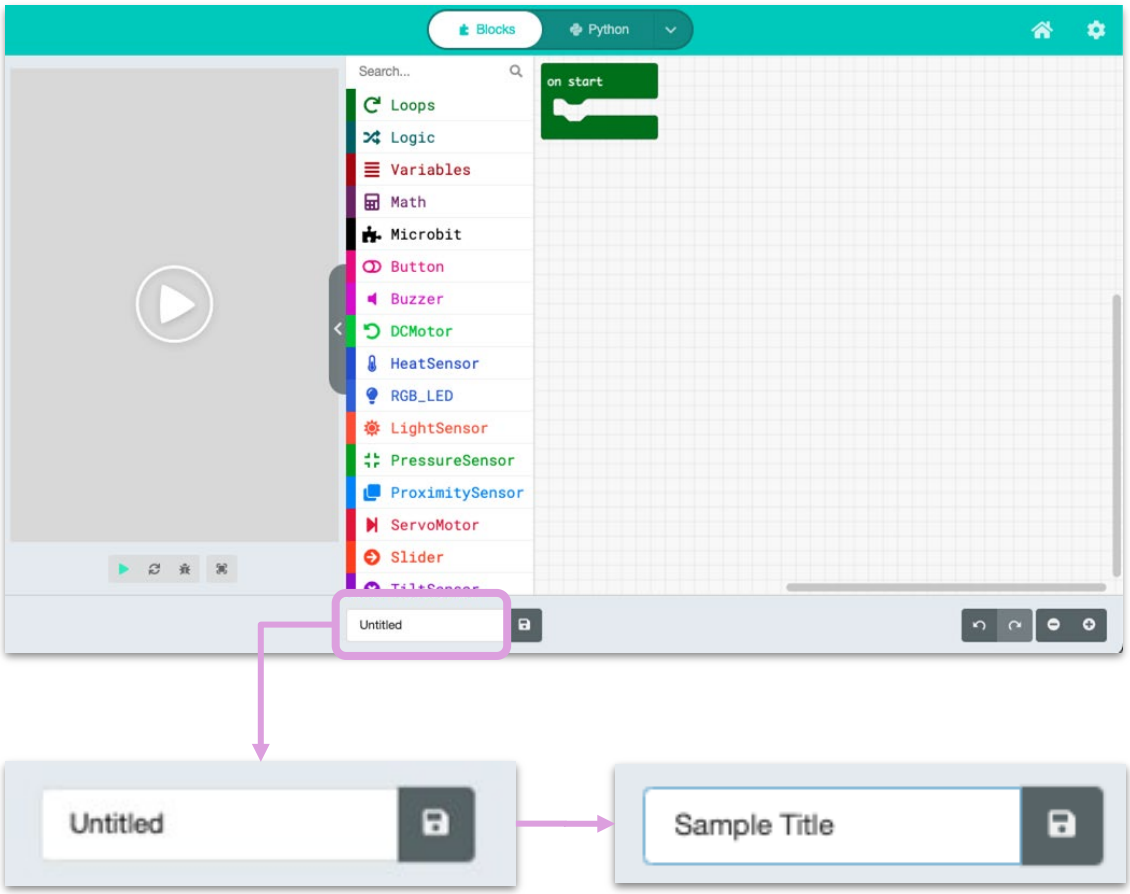
點擊下拉選單以在Script選項（JavaScript 和 Python）之間切換。

現在您已經準備好開始編程了！

在SAM Script中
新增檔名&儲存專案

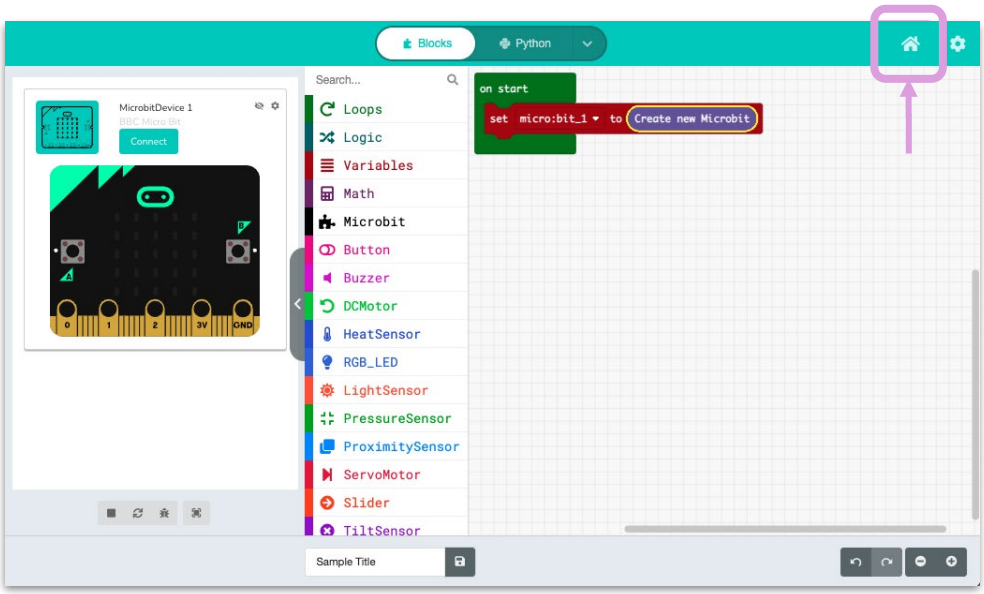
1. 當您在一個專案中時，您可以點擊 “Untitled”(未命名)欄位來為您的專案建立檔名。這個檔名將自動儲存，當您回到主頁時會顯示在專案檔案圖示上。

   檔名欄位在畫面底部



在SAM Script中
新增檔名&儲存專案

2. 當您完成編程後，請點擊右上角的“Home”(首頁)按鈕圖示。主頁將顯示您的專案並顯示新的檔名。



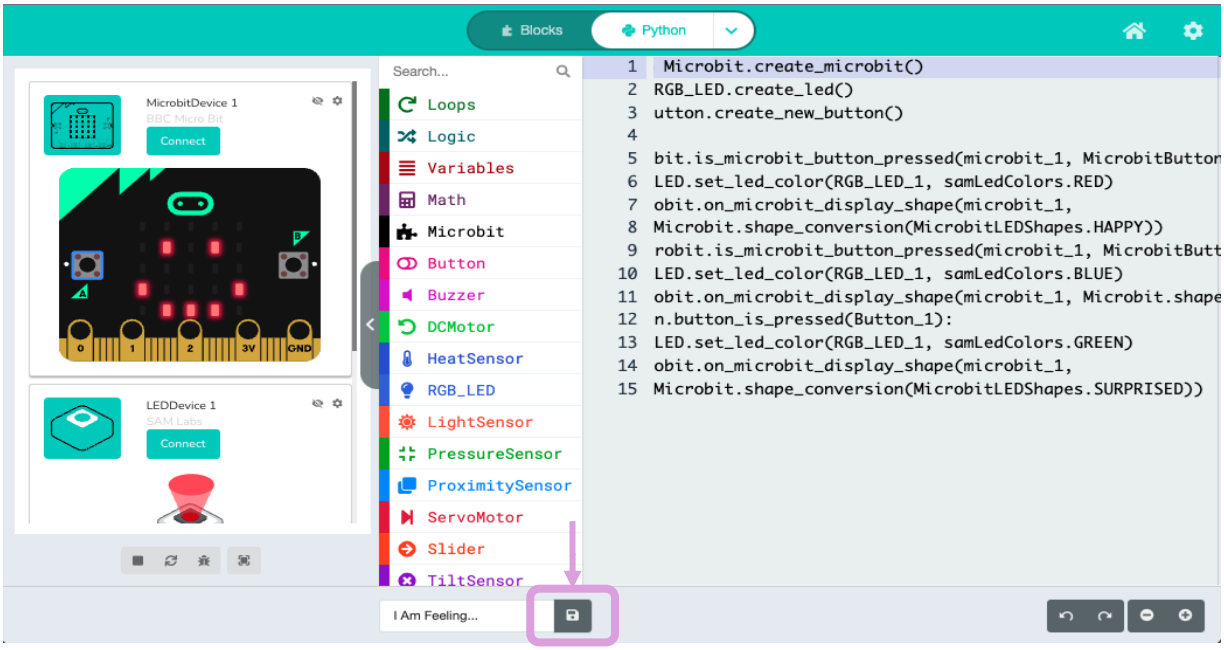
*注意，專案左上方的圖示顯示了專案最後是使用哪種編程語言編寫的：

Block　Python　JavaScript

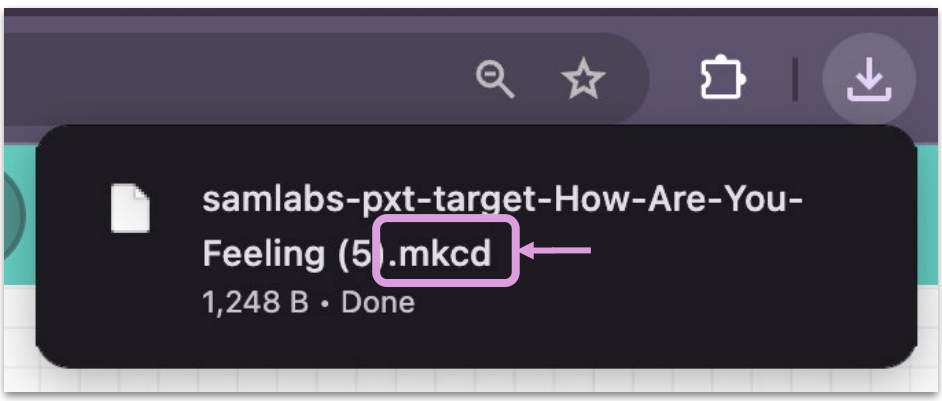
在SAM Script中
下載&匯入專案

下載專案

1. 當您在一個想要下載到您的裝置的專案內時，請點擊專案標題旁邊的“Save” (儲存)圖示：



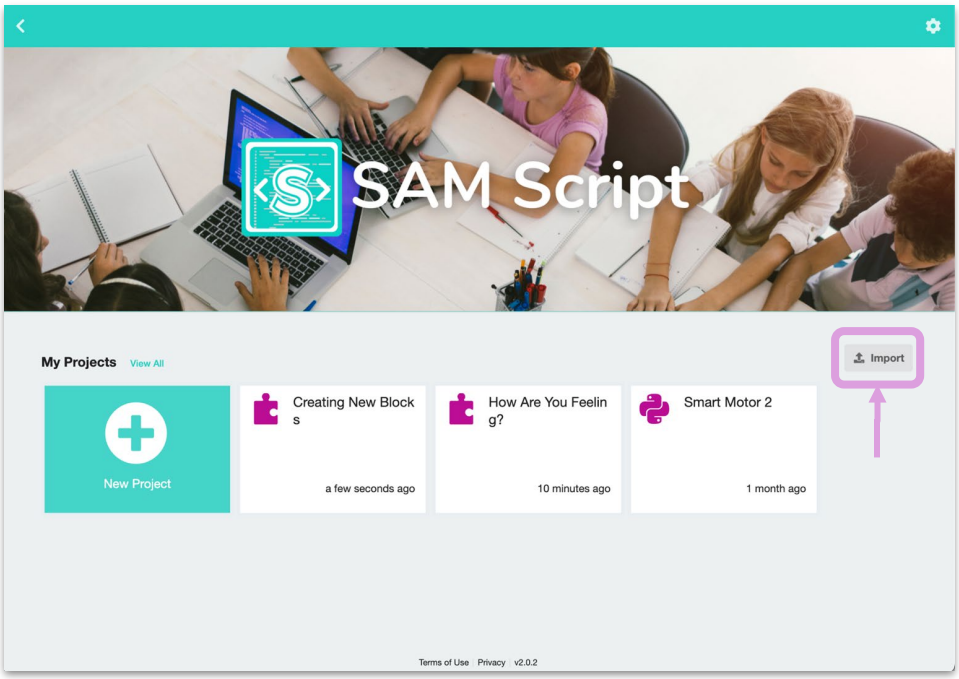
2. 這個專案將自動下載到您的裝置上。請注意，SAM Script 專案的檔案類型是以“.mkcd”的格式儲存。



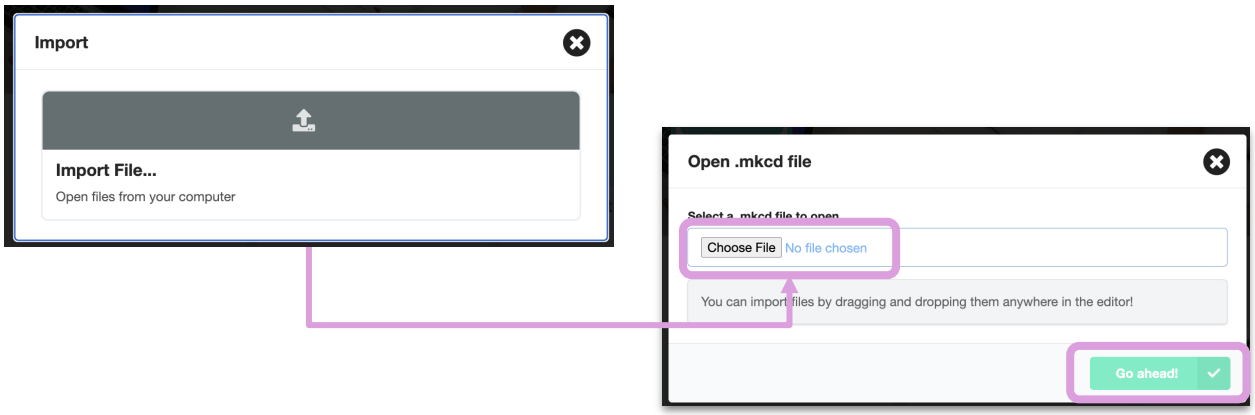
在SAM Script中
下載&匯入專案

匯入專案

1. 在 SAM Script 中，請點擊主畫面右上方的“Import” (匯入)



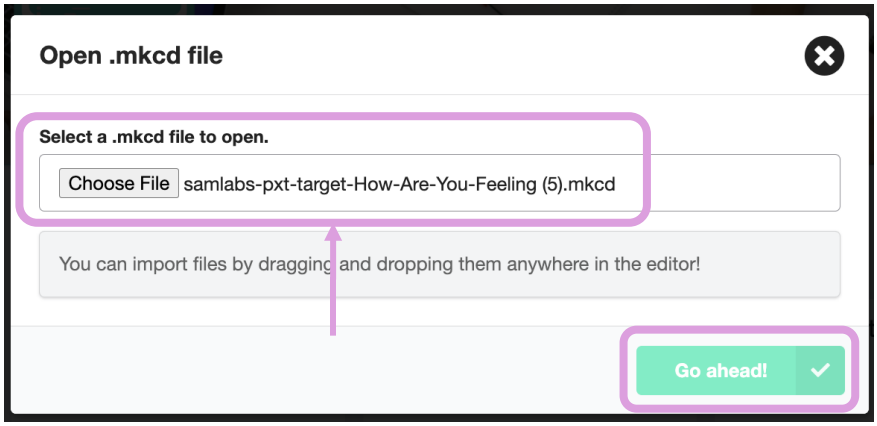
2. 一個“Import” (匯入)畫面將允許您選擇從裝置中儲存的 .mkcd 專案檔案。



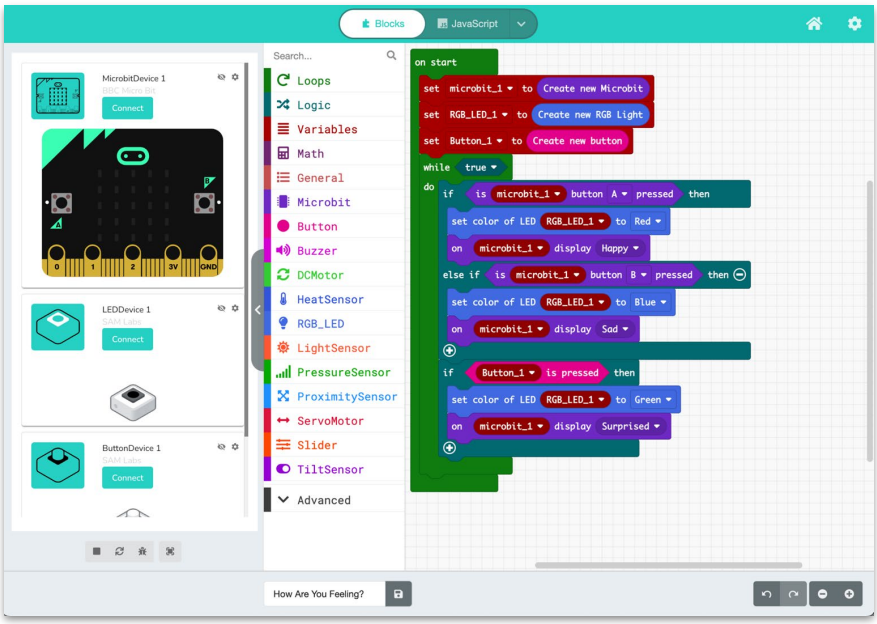
在SAM Script中
下載&匯入專案

匯入專案

3. 您將在欄位中看到檔案的標題。點擊“Go ahead”(繼續)以匯入該檔案。



4. 您將看到專案已開啟，並且現在已經儲存在您的主畫面上。

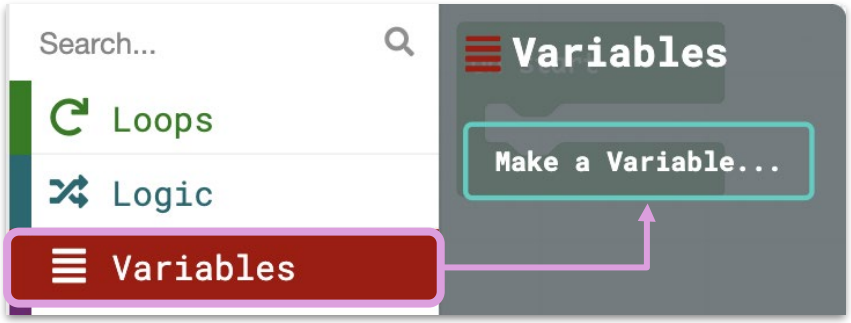


新增虛擬模塊

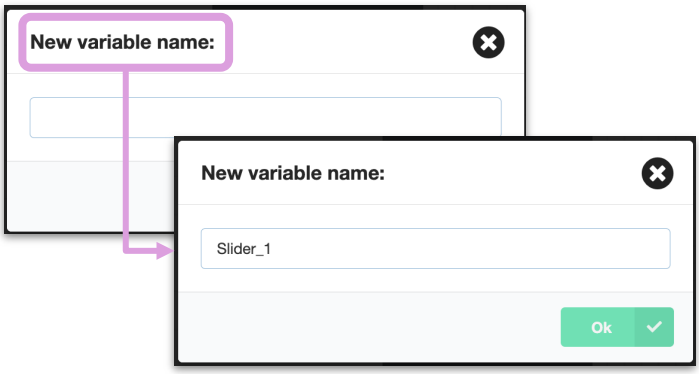
在 SAM Script 專案中新增 SAM 模塊和 micro:bit 與在 SAM Blockly 中的步驟有所不同。以下的步驟更符合軟體工程師的邏輯，將學習從 SAM Blockly 的結構化體驗過渡到更真實的環境，以協助學生為高中及更高層次的學習做好準備。

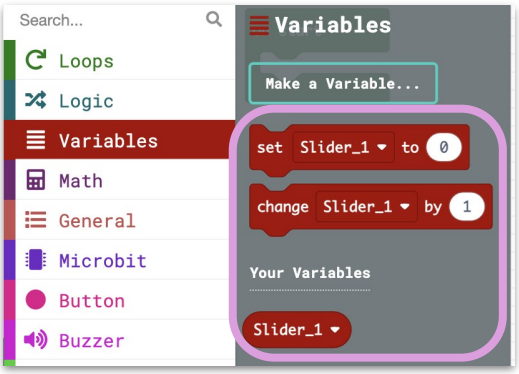
在 block 語言新增 SAM 模塊 & micro:bits

1. 在您的新專案中，點擊“**Variables**”(變數)選單，然後選擇“Make a Variable...”(建立變數)



2. 為您的變數取一個您想要新增的SAM 模塊 或的名稱。這可以是任何名稱，但為了方便起見，最好將其命名為 snake case (underscores for spaces)(蛇行命名法) ，如此範例所示。



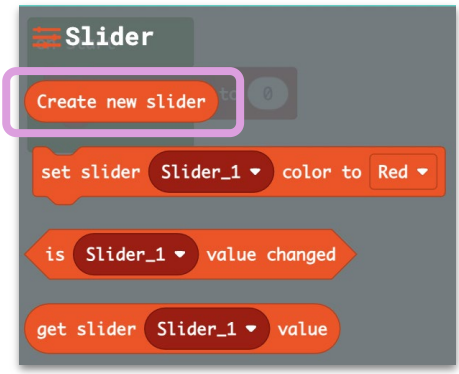


您的新變數將顯示在選單中。
這個範例展示了如何將一個Slider(滑軸)加入程式中。

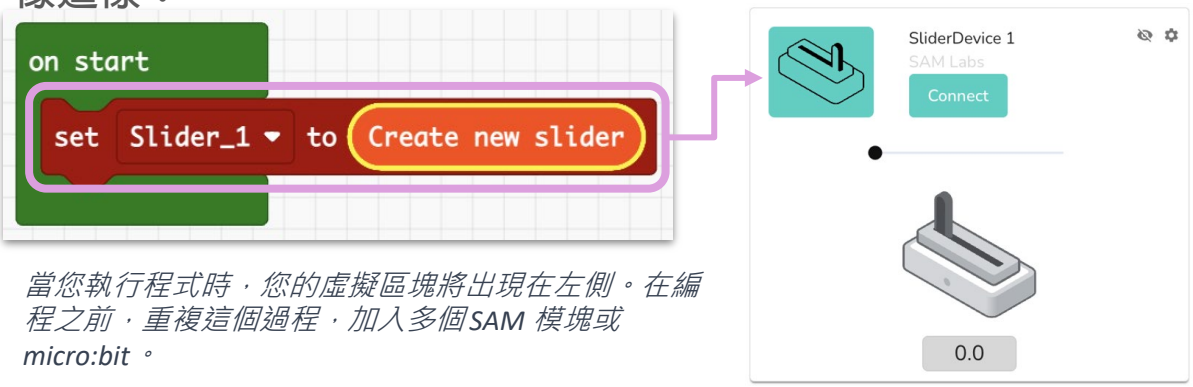
新增虛擬模塊

3. 從“**Variables**” (變數)選單中，選擇“**Set [variable name] to 0**”(將[變數名稱] 設為 0)積木，然後將其加入 “on start” (在開始時)的積木。

4. 現在，轉到您正在配對的 SAM 模塊或 micro:bit 的選單。這個範例需要使用**“Slider” menu. (滑軸選單)**頂部的積木選項將是“Create new [block].” (建立新的模塊)



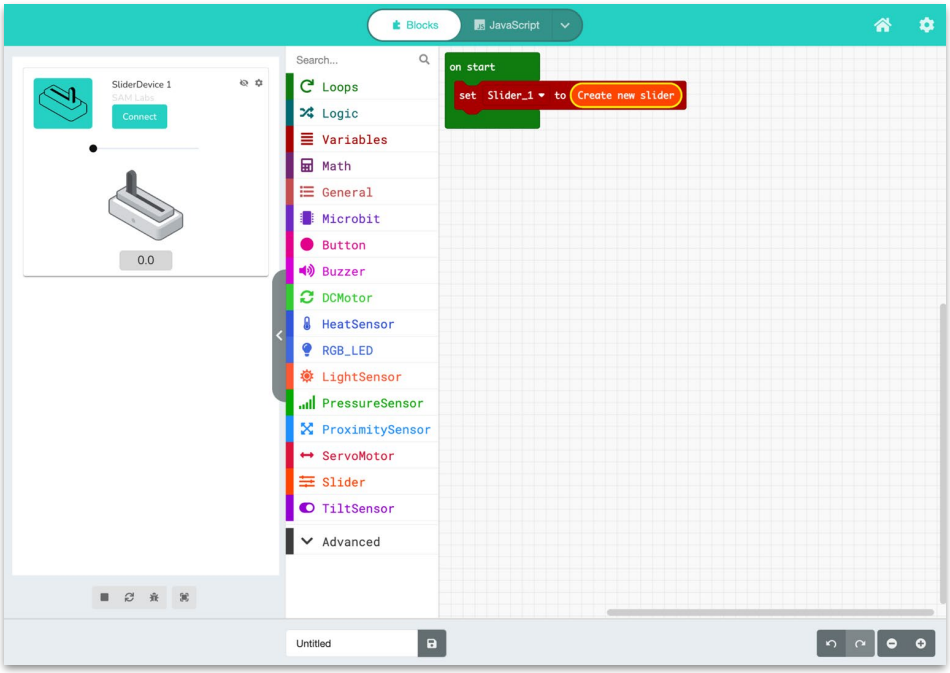
5. 拖曳“Create new [block]” (建立新的模塊)並將其連接到“set [variable] to 0,” (將變數設為0)的內部，取代原本的 0。它將看起來像這樣：



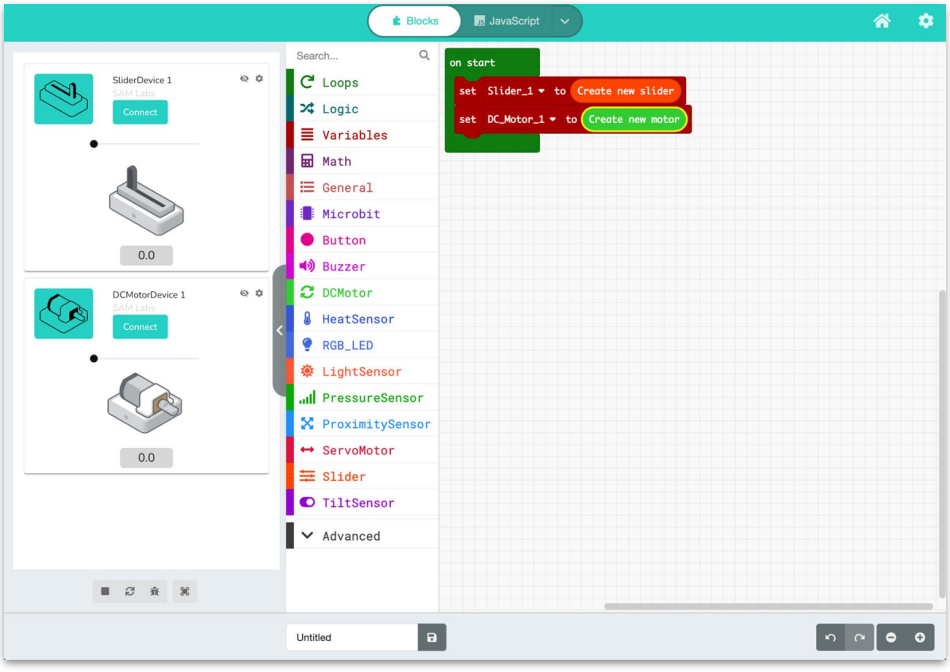
當您執行程式時，您的虛擬區塊將出現在左側。在編程之前，重複這個過程，加入多個SAM 模塊或 micro:bit 。

新增虛擬模塊

您的最終畫面將呈現如下：



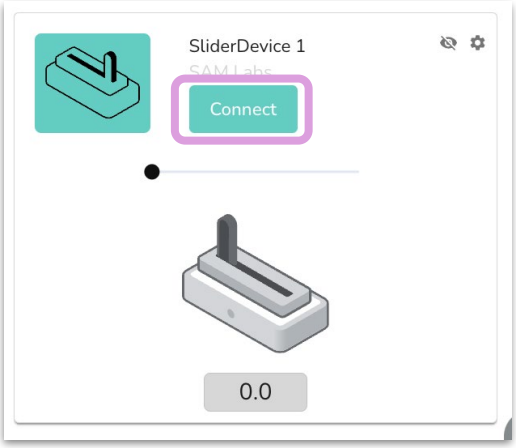
搭配滑軸和直流馬達的範例：



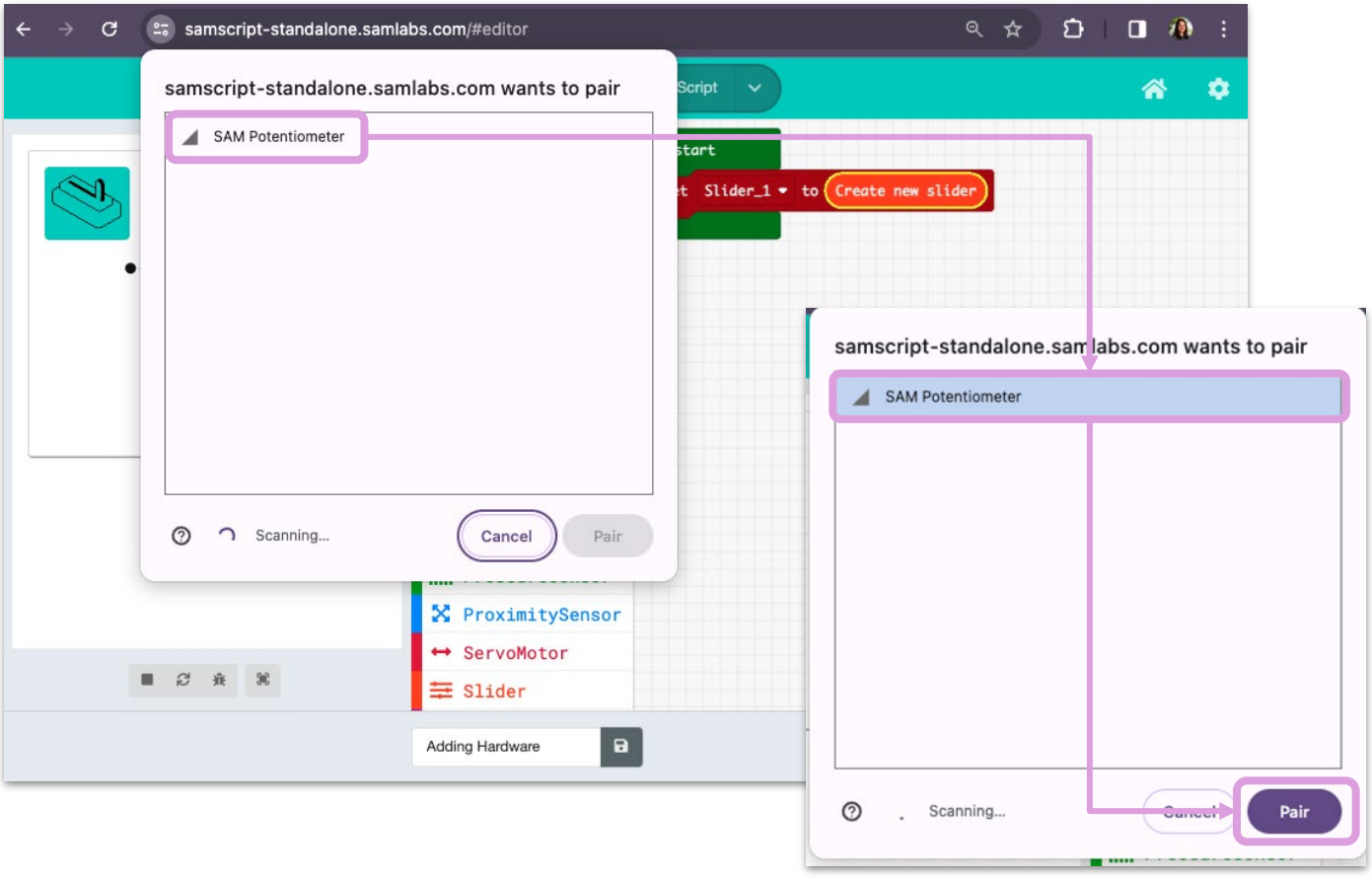
透過藍牙新增硬體

1. 在左側的虛擬裝置面板中，點擊 “Connect.” (連線)

   確保您的 SAM 模塊已經開啟。

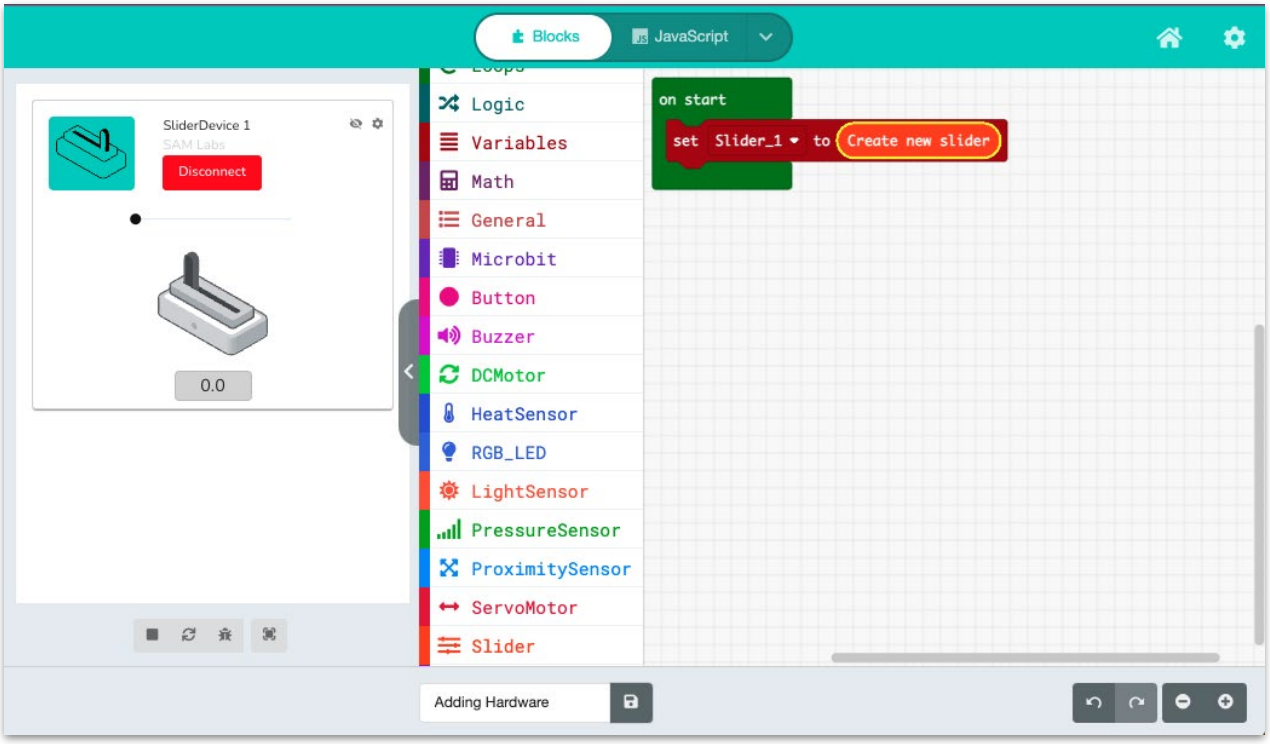


2. 您的模塊或 micro:bit 會在 Google 彈出視窗中顯示，就像在 SAM Blockly 中一樣。選擇該裝置，然後點擊“Pair.”(配對)

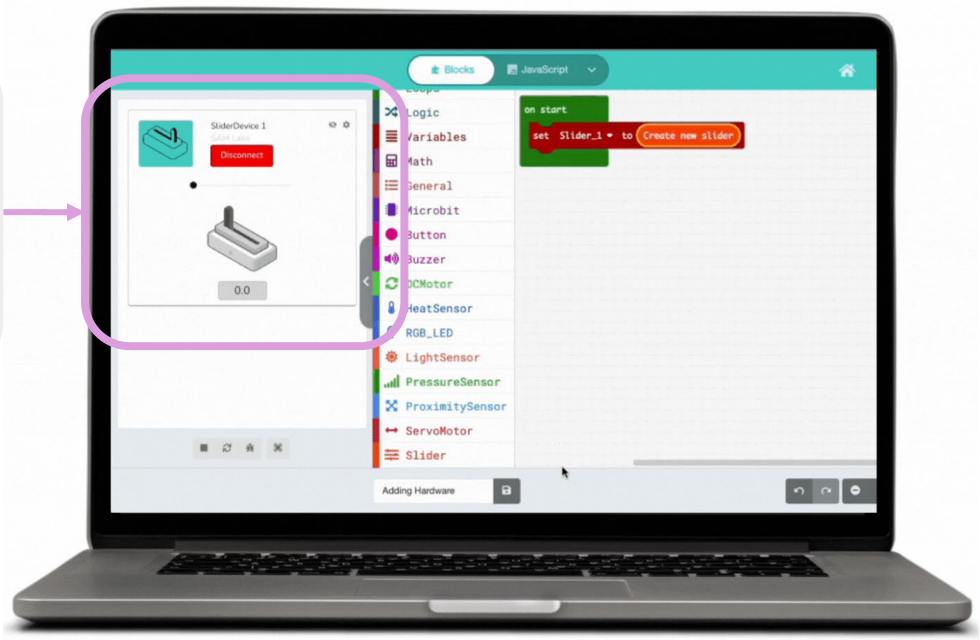


透過藍牙新增硬體

3. 您的裝置現在已經成功配對。試著測試它，看看在虛擬版本中的實際運作。若要斷開硬體裝置的連線，請在左側面板中點擊 “**Disconnect**” (中斷連線)。



看看這裡，一旦實體滑軸成功配對，就能控制螢幕上的虛擬滑軸。



新增虛擬模塊

在JavaScript中新增SAM 模塊 & micro:bits

1. 在您的新專案中，請點擊“JavaScript”選單。您將透過在第一行輸入“let”來建立一個變數。

2. 在空格後，輸入變數的名稱（使用蛇形命名法，每個空格使用底線）。

以下是如果您要將所有裝置都加入程式中時，使用的變數的預設名稱範例清單：

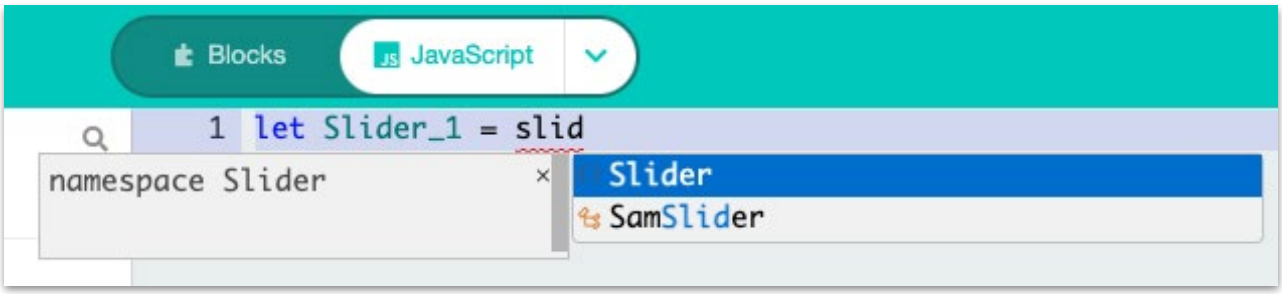
```
1  let Microbit_1
2  let Button_1
3  let Buzzer_1
4  let DC_Motor_1
5  let Heat_Sensor_1
6  let RGB_LED_1
7  let Light_Sensor_1
8  let Pressure_Sensor_1
9  let Proximity_Sensor_1
10 let Servo_Motor_1
11 let Slider_1
12 let Tilt_Sensor_1
```

新增虛擬模塊

在 JavaScript中新增SAM 模塊 & micro:bits

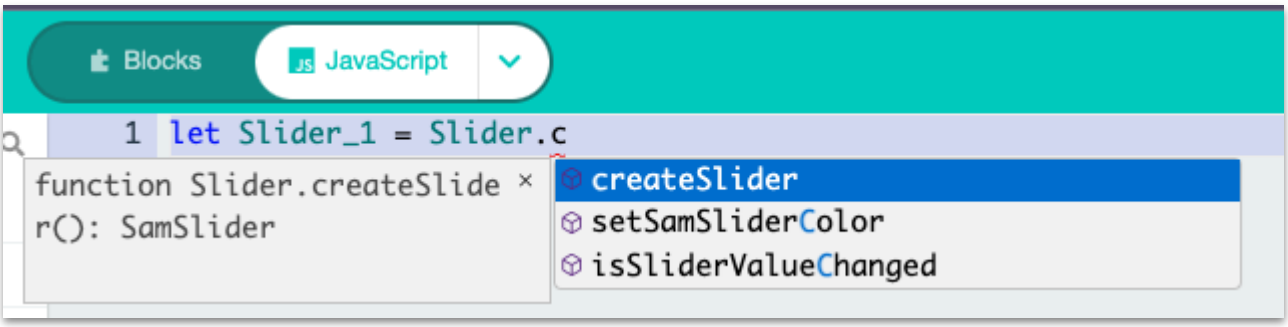
3. 在空格後，輸入“ = [name of block]” (= [模塊名稱]) 。當您開始輸入模塊名稱時，將會出現一個選單。點擊該模塊的名稱。

   在這個範例中，您會看到“Slider” (滑軸)出現。



4. 在空格後，輸入“ .create” 。當您開始輸入時，將會出現一個選單，其中包含建立該模塊的選項。選擇該選項。

   在這個範例中，它是“createSlider”



新增虛擬模塊

在 JavaScript中新增SAM 模塊 & micro:bits

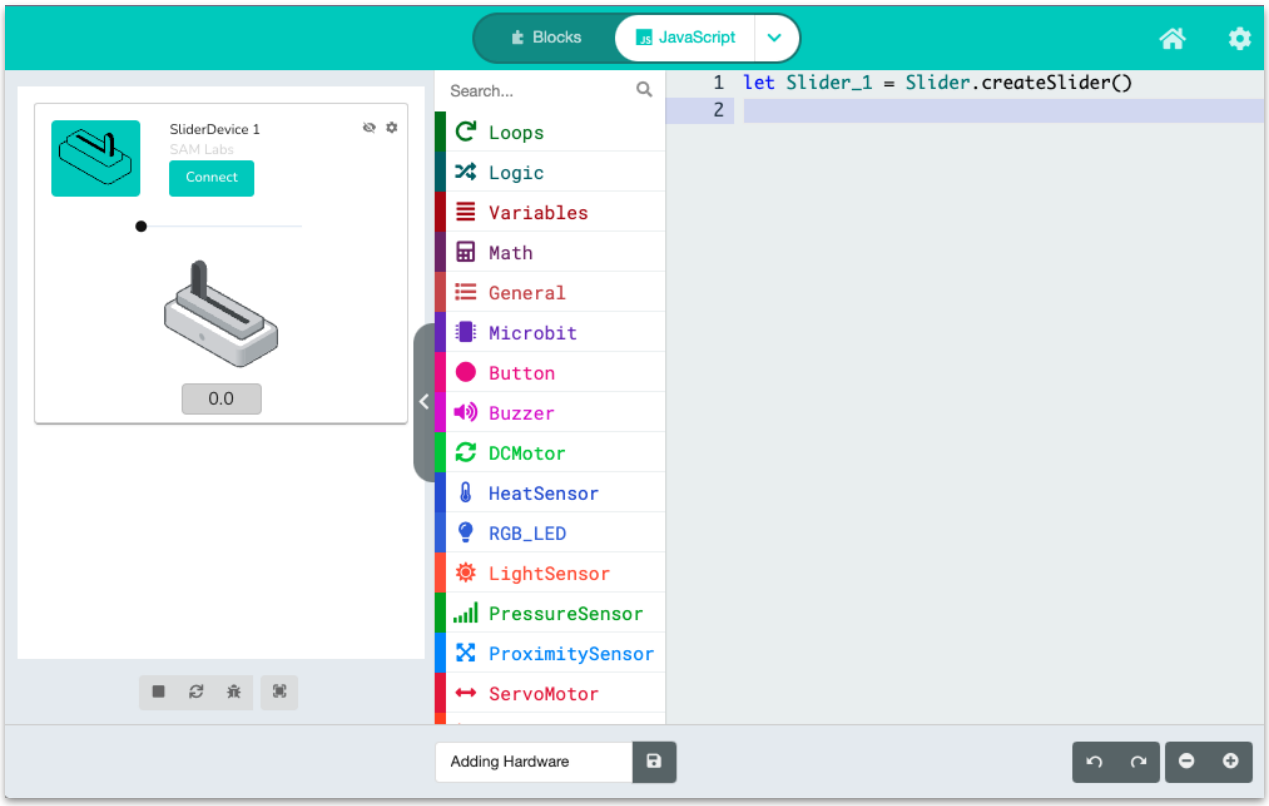
5. 不加空格，加入“()”

   在這個範例中，最後一行應該是這樣的：

```
1  let Slider_1 = Slider.createSlider()
2
```

   這一行程式碼已經完成，您可以重複這個步驟，將任何其他 SAM 模塊或 micro:bits加入到程式中。

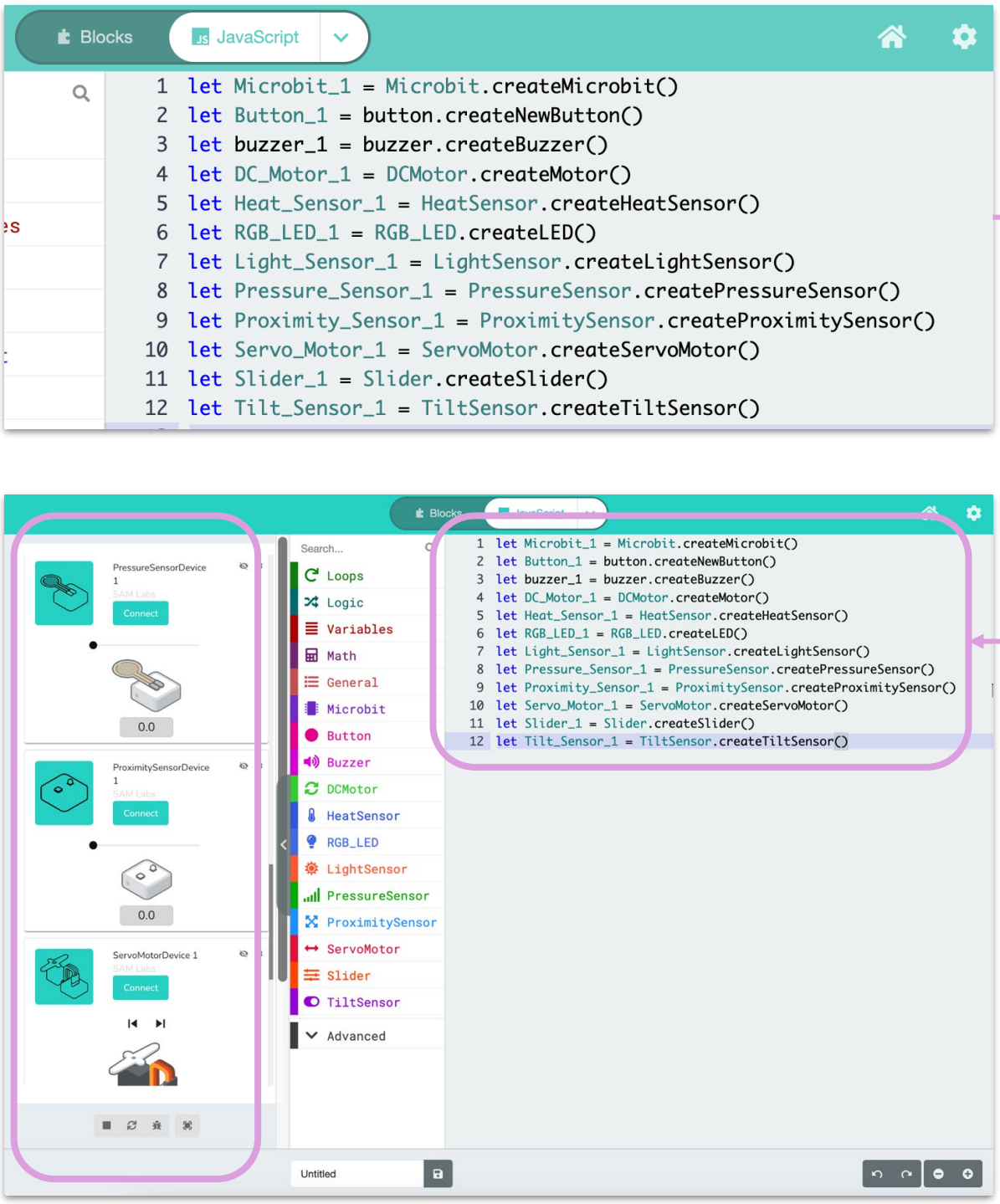
6. 透過執行程式進行測試。您應該會看到虛擬模塊出現。連接硬體需要在第 14-15頁上採取相同的步驟。



新增虛擬模塊

在 JavaScript中新增SAM 模塊 & micro:bits

以下是一個範例清單，顯示在一個程式中使用 JavaScript 加入的所有 12 個裝置，以及左側面板中相應的裝置：

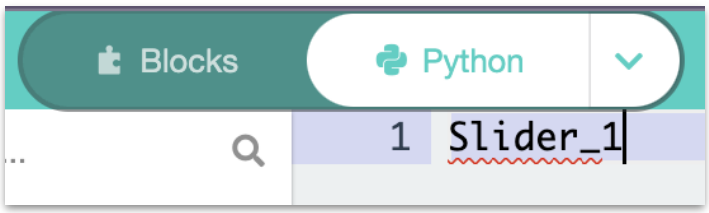


新增虛擬模塊

在 Python 中新增 SAM 模塊 & micro:bits

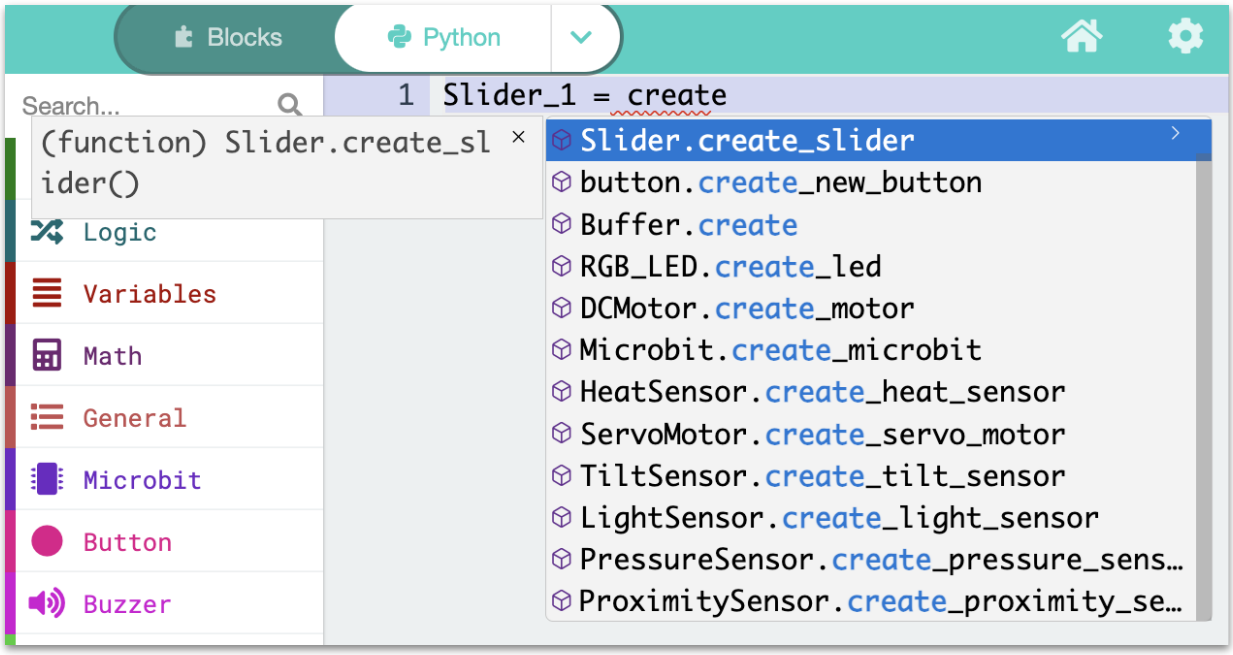
1. 在您的新專案中，請點擊“Python”選單。您將透過輸入變數的名稱（使用蛇形命名法，每個空格使用底線）來建立一個變數。

   在這個滑軸的範例中，名稱是 “Slider_1”



2. 在空格後，輸入“ = create。當您開始輸入 ‘create這個詞時，將會出現一個選單。點擊積木的名稱block.create_block。

   在這個範例中，點擊“Slider.create_slider”

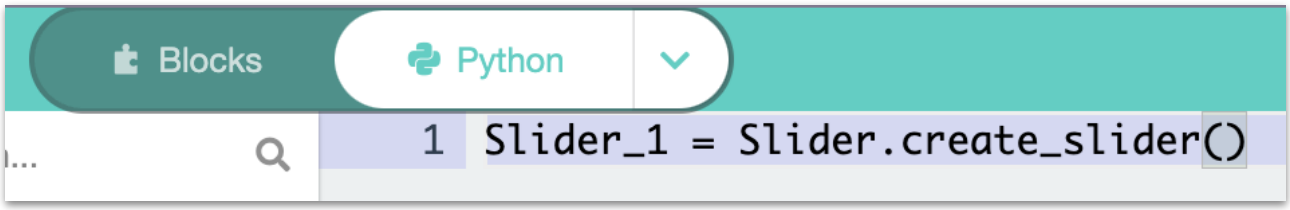


新增虛擬模塊

在 Python 中新增 SAM 模塊 & micro:bits

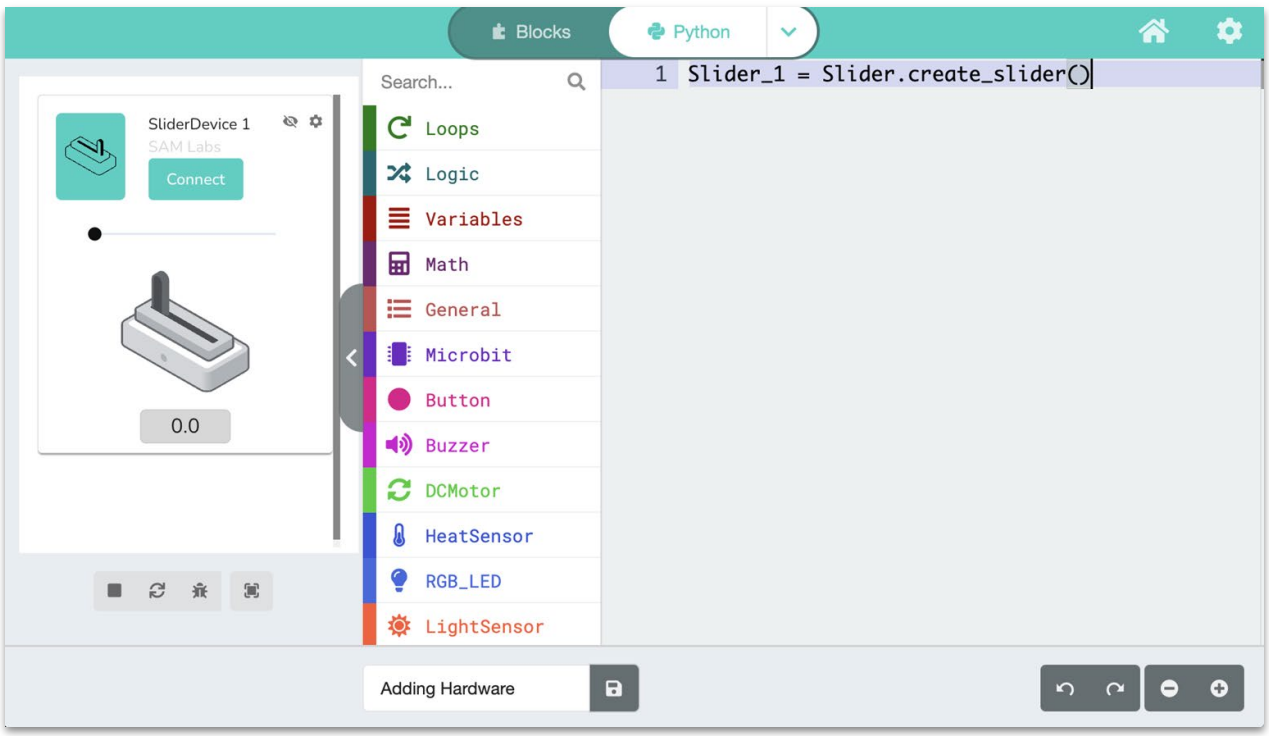
3. 在您從選單中選擇正確的“create”選項以符合您的模塊後，您將看到“()”會自動出現在該行的末尾。

   在這個範例中，最後一行應該是這樣的：



   這一行程式碼已經完成，您可以重複這個步驟，將任何其他 SAM 模塊或 micro:bit 加入到程式中。

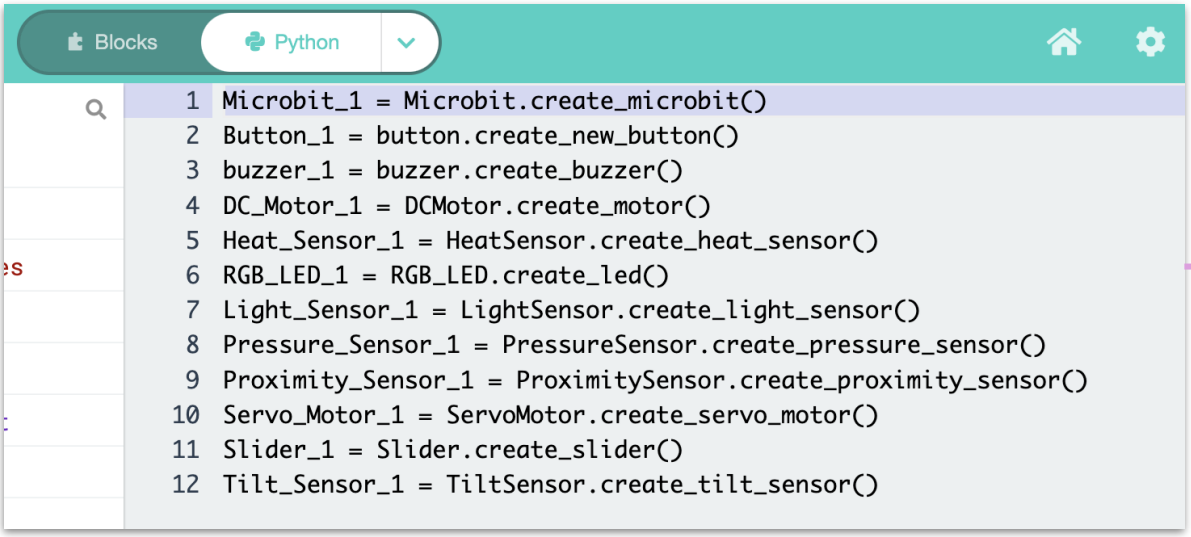
4. 透過執行程式進行測試。您應該會看到虛擬模塊出現。連接硬體需要在第 14-15頁上採取相同的步驟。

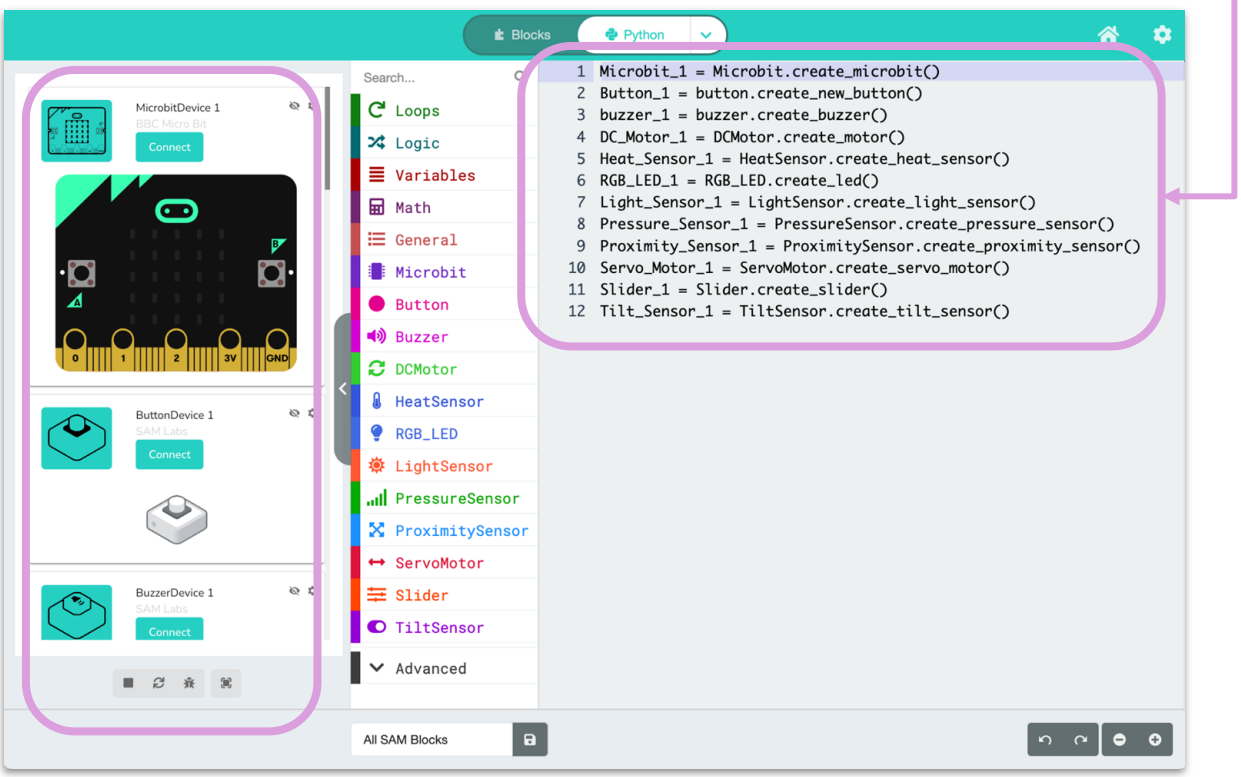


新增虛擬模塊

在 Python 中新增 SAM 模塊 & micro:bits

以下是在一個程式中使用 Python 加入的所有 12 個裝置的範例清單，以及左側面板中相應的裝置：



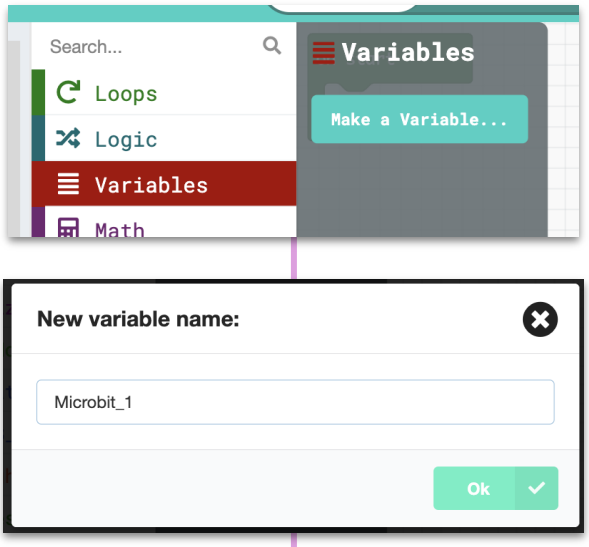


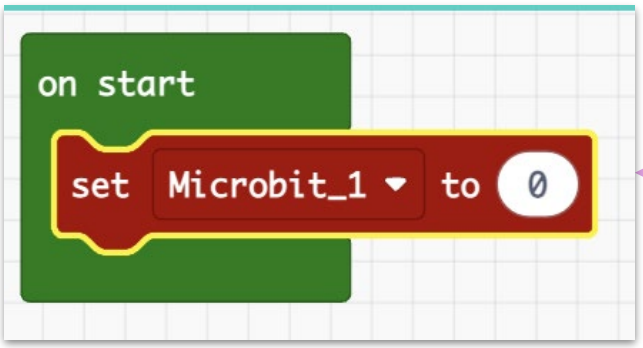
讓我們開始：基本入門程式碼

接下來的部分是為了提供一些簡單的程式碼，讓您更瞭解這個平台。這將包含Block、JavaScript、Python 以及各種不同的裝置。請按照步驟建立這些範例專案，以便與您的使用者和潛在使用者分享。

練習程式碼 1: “數位名牌”

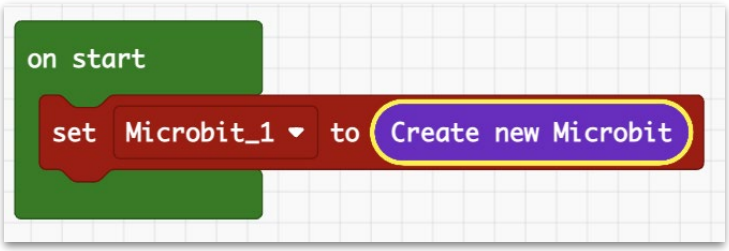
1. 開啟一個新專案。透過建立一個變數，將一個 micro:bit 加入：
   a. **Variable** 選單中
   b. “**Make a Variable...**”(建立變數)
   c. 命名為 “**Microbit_1**”
   d. 將 “**set Microbit_1 to 0**” (將Microbit_1設為0)拖曳到to “**on start**.”(在開始時)



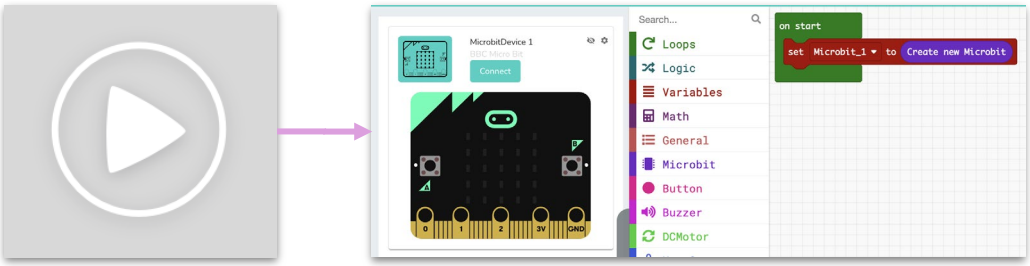


練習程式碼 1:
“數位名牌”

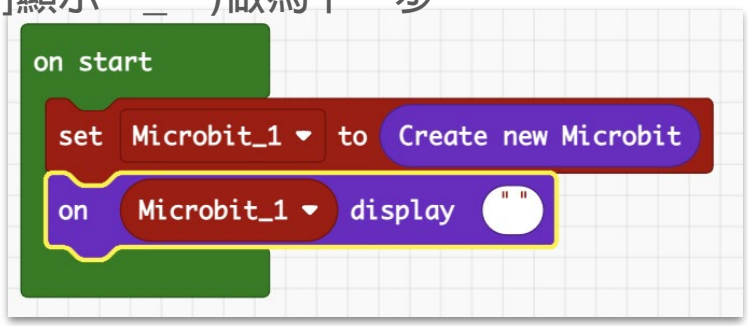
2. 從 “Microbit” 選單中，將 “Create new Microbit” (建立新的Microbit) 拖曳到 0 上。



3. 測試程式。按下播放按鈕，查看虛擬 micro:bit。

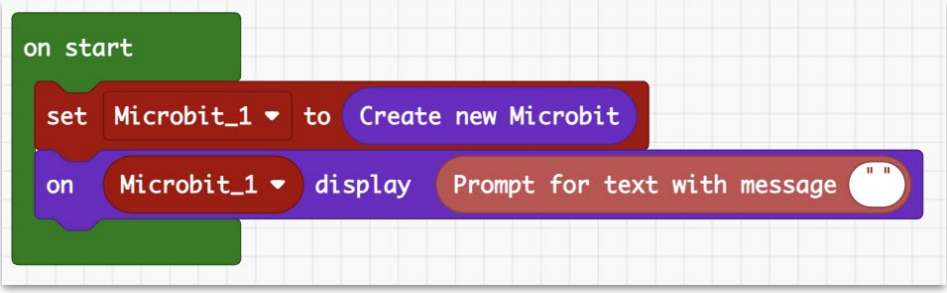


3. 從 “Microbit” 選單中，新增 “on [Microbit_1] display “_” ” (在 [Microbit_1]顯示 “_”)做為下一步

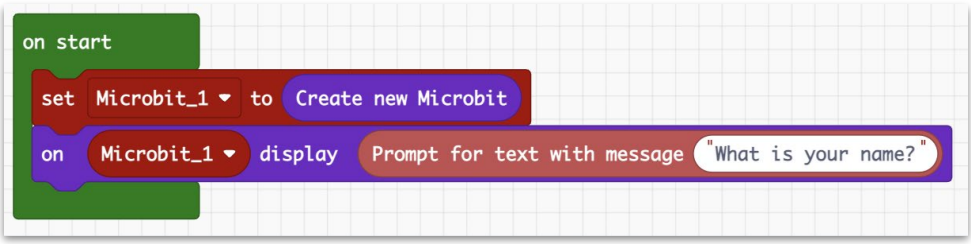


練習程式碼 1:
“數位名牌”

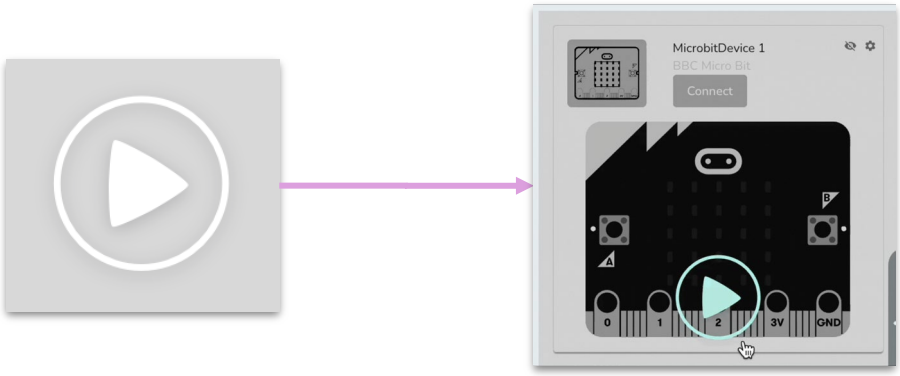
4. 從 “General”(一般)選單中，新增 “Prompt for text with message “_” ”(使用訊息” _” 提示文字)作為下一步。



5. 在提示方塊中輸入問題 “*What is your name?*” (你的名字是?)

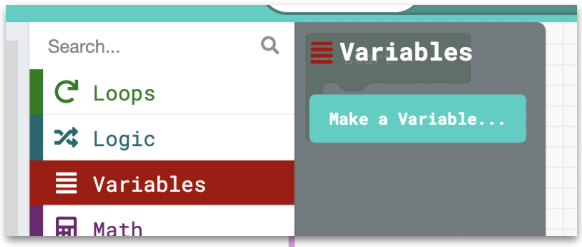


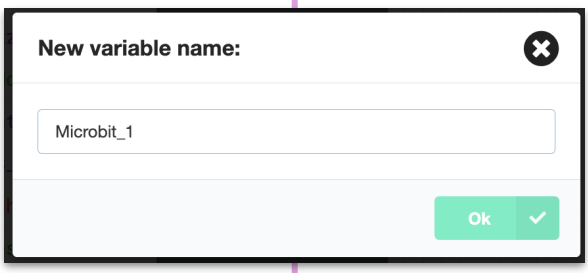
6. 測試程式。按下執行按鈕，查看程式。

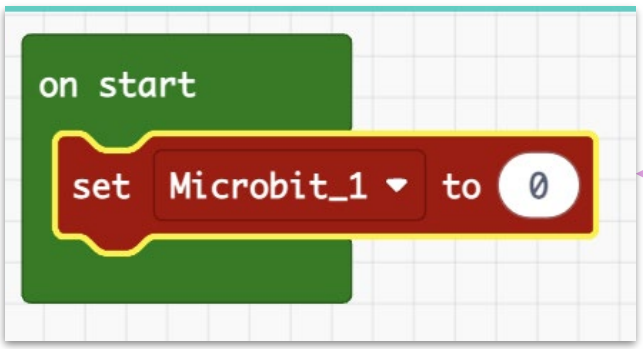


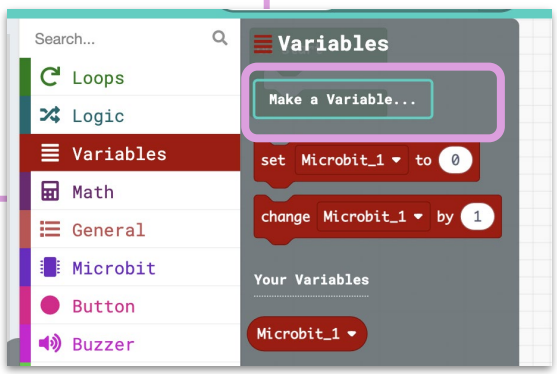
練習程式碼 2:
“跳動的心”

1. 開啟一個新專案。透過建立一個變數，將一個 micro:bit 加入：
   a. Variable 選單中
   b. “Make a Variable...”(建立變數)
   c. 命名為 “Microbit_1”
   d. 將 “set Microbit_1 to 0” (將Microbit_1設為0)拖曳到to “on start.”(在開始時)
   e. 從 “Microbit” 選單中，將 “Create new Microbit”拖曳到 0。

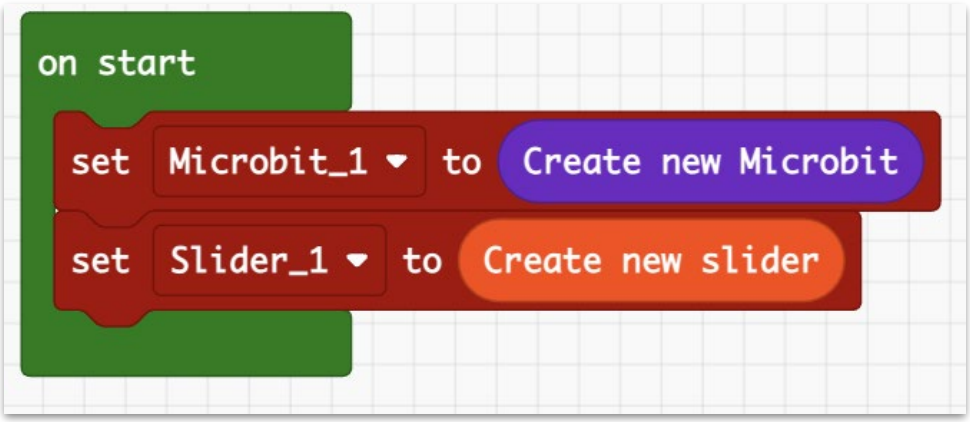






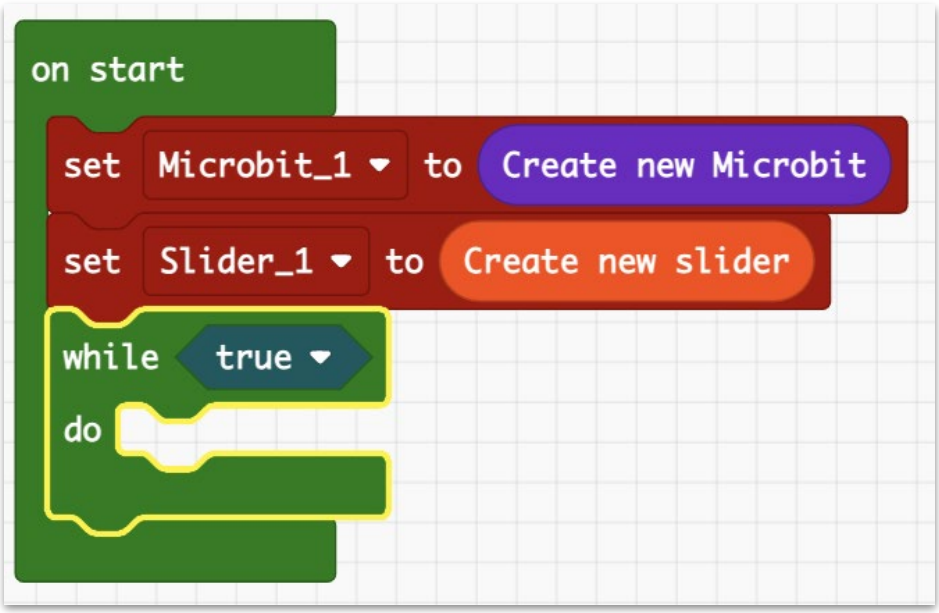


重複相同步驟，新增一個滑軸。從“Slider”(滑軸)選項中，將 “Create new Slider” (建立新的滑軸)拖到 0 上。

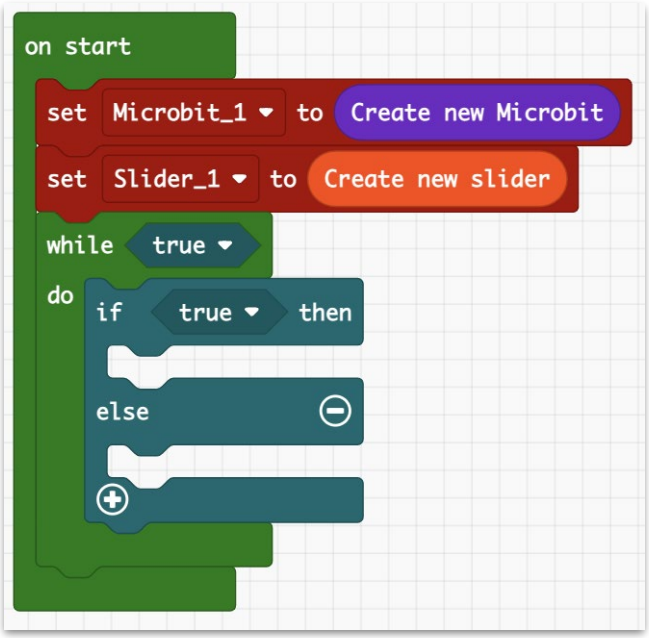


練習程式碼 2:
“跳動的心”

2. 從 “**Loops**”(迴圈)選項中，拖曳一個 “**while, do**” 迴圈做為下一步，並設定為“**true**.”

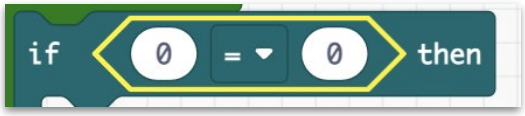


3. 從 “**Logic**”(邏輯)選項中，拖曳一個 “**if, do, else**” 並放在迴圈內。

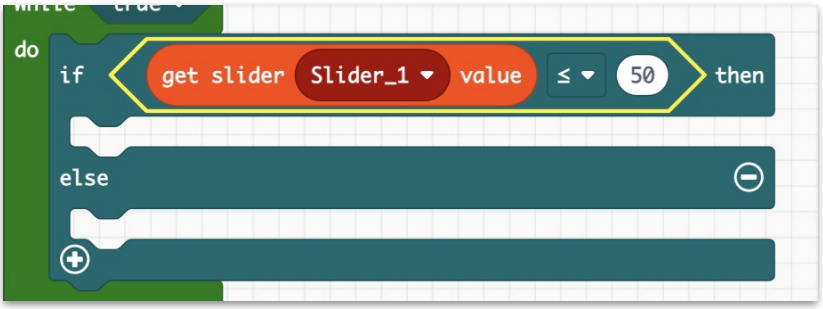


練習程式碼 2:
“跳動的心”

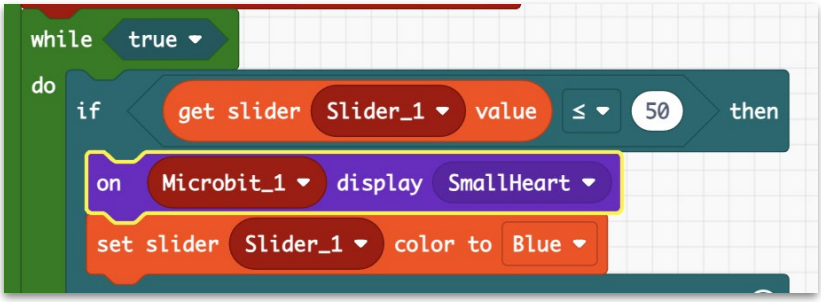
4. 從 “Logic”(邏輯)選項中，
   a. 在比較區塊中拖曳出 “if, do, else.”



   a. 從 “Slider”(滑軸)選項中，新增 “get slider Slider_1 value”(取得滑軸Slidr_1的值)作為第一個數值
   b. 設定符號為“≤”
   c. 設定數字為“50”

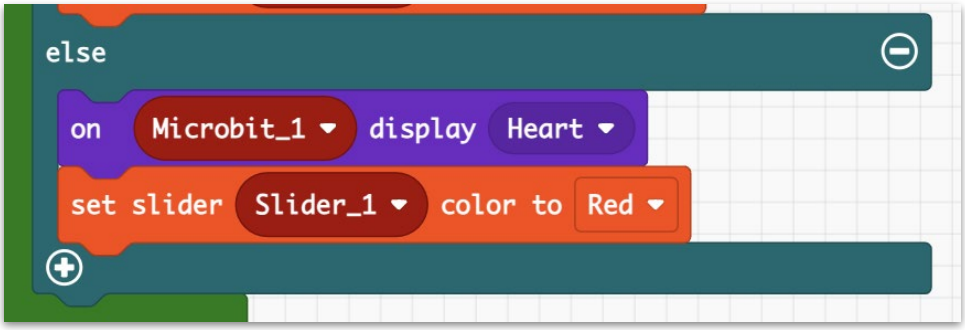


5. 在“if”(如果)的部分：
   a. 從 “Microbit” 選單中，拖曳 “on Microbit_1 display” (在Microbit_1顯示)，然後新增 “SmallHeart” 作為圖像。
   b. 從 “Slider”(滑軸)選單中，新增 “set slider Slider_1 color to”(將滑軸Slider_1顏色設定為)，然後將顏色設定為 “Blue.” (藍色)。

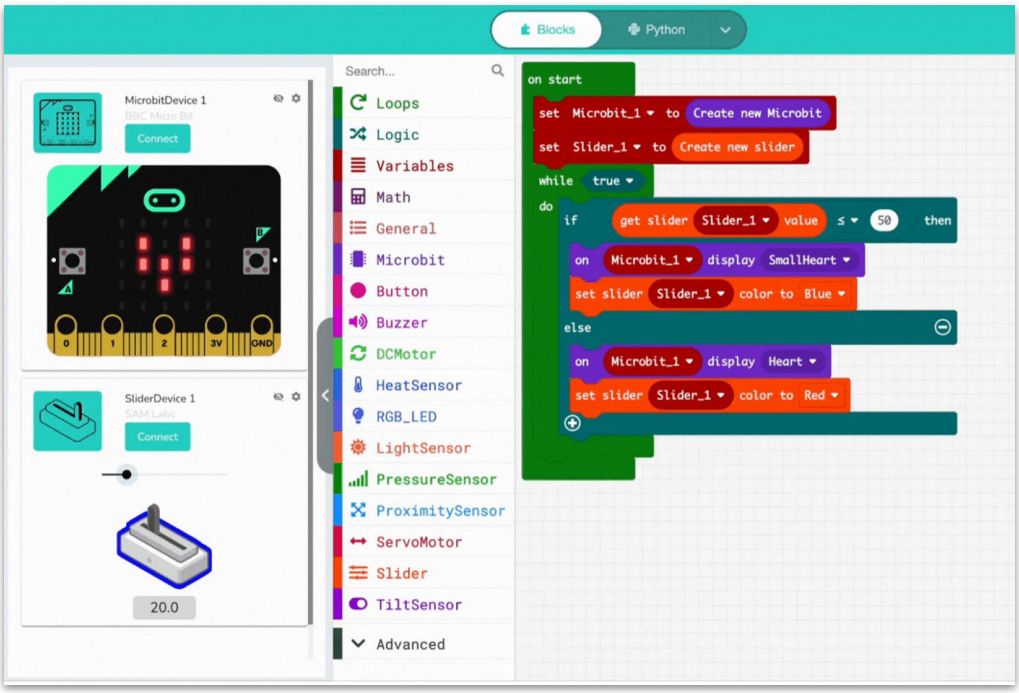


練習程式碼 2:
“跳動的心”

6. 在“else”(否則)的部分：
   a. 從 “Microbit” 選項中，拖曳“on Microbit_1 display”(在Microbit_1顯示)，然後新增“Heart”(愛心)作為圖像。
   b. 從 “Slider”(滑軸)選項中，新增 “set slider Slider_1 color to”(將滑軸Slider_1顏色設定為)，然後將顏色設定為“Red.”(紅色)。

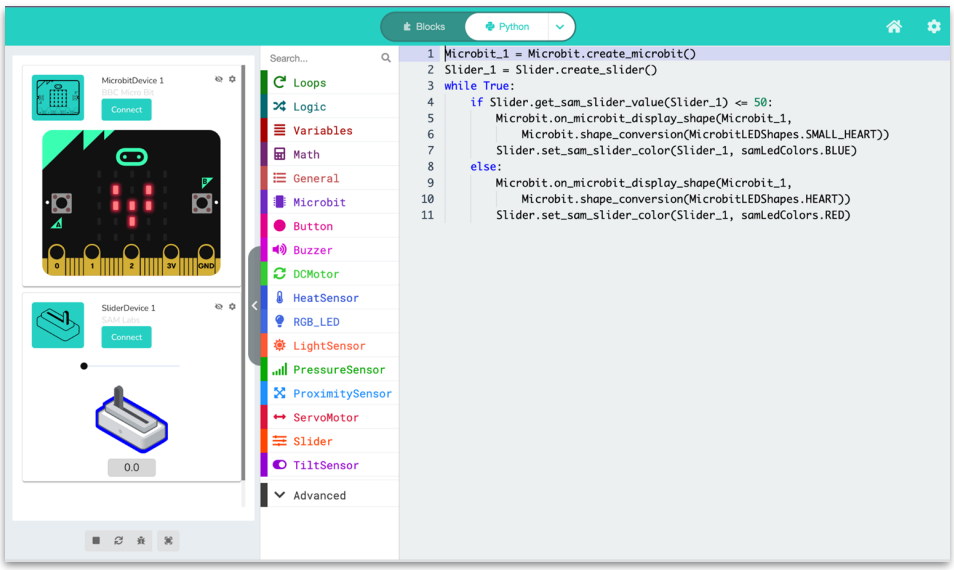


7. 測試程式。按下執行按鈕，查看程式。使用滑桿來「擠壓」心臟，觀察基於滑桿數值的紅色和藍色變化。

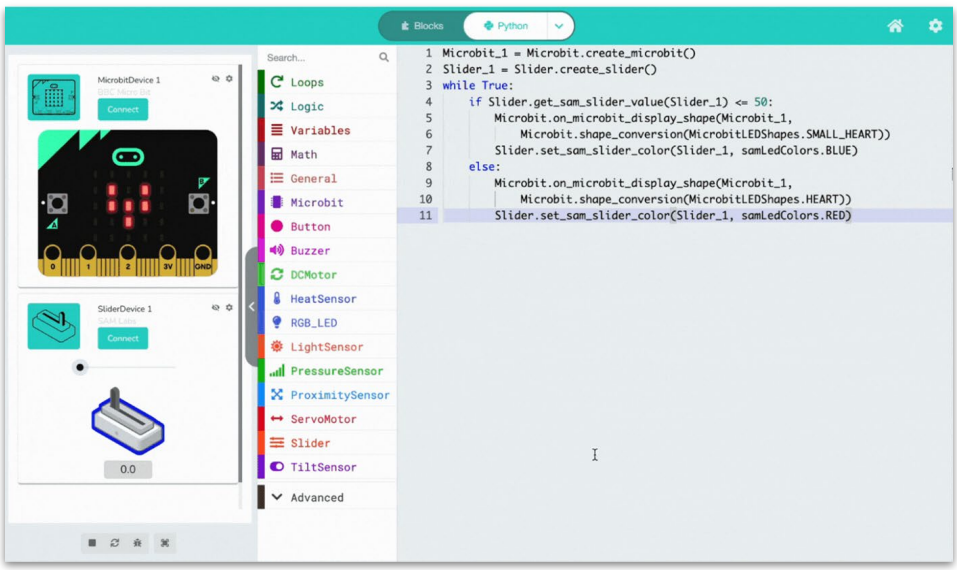


練習程式碼 2:
“跳動的心”

8. 選擇頂部選單中的 Python 以進行轉換：

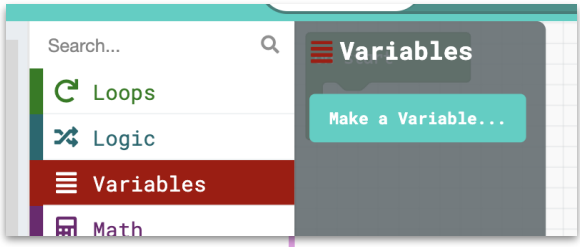


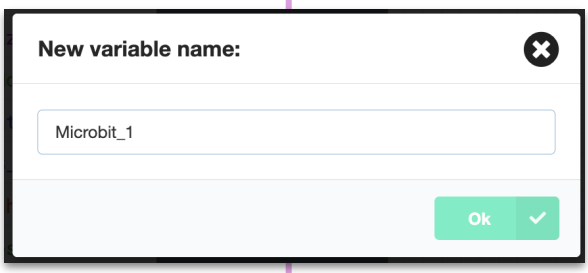
7. 在“**if, do, else**” 區塊中，將 “RED” (紅色)改為 “BLUE”(藍色)，將 “BLUE” (藍色)改為 “RED”(紅色)，然後重新測試程式。您應該會看到新的條件是，當數值介於 0 到 50 之間時，滑桿變為藍色；當數值介於 51 到 100 之間時，滑桿變為紅色。

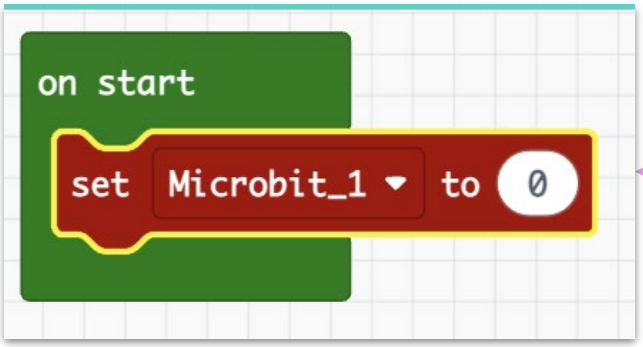


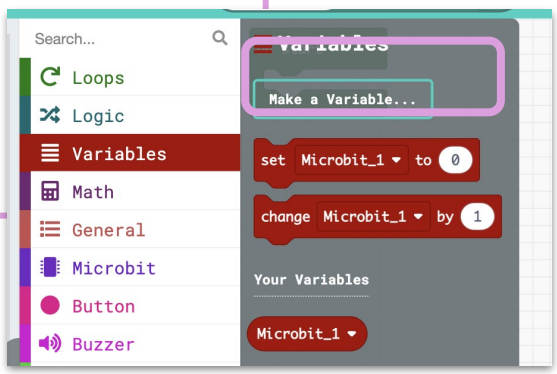
練習程式碼 3:
“奇怪知識驗證器”
(第一部分)

1. 開啟一個新專案。透過建立一個變數，將一個 micro:bit 加入：
   a. **Variable** 選單中
   b. “**Make a Variable...**”(建立變數)
   c. 命名為 “**Microbit_1**”
   d. 將 “**set Microbit_1 to 0**” (將Microbit_1設為0)拖曳到to “**on start**.”(在開始時)
   e. 從 “**Microbit**” 選單中，將 “**Create new Microbit**”拖曳到 0。

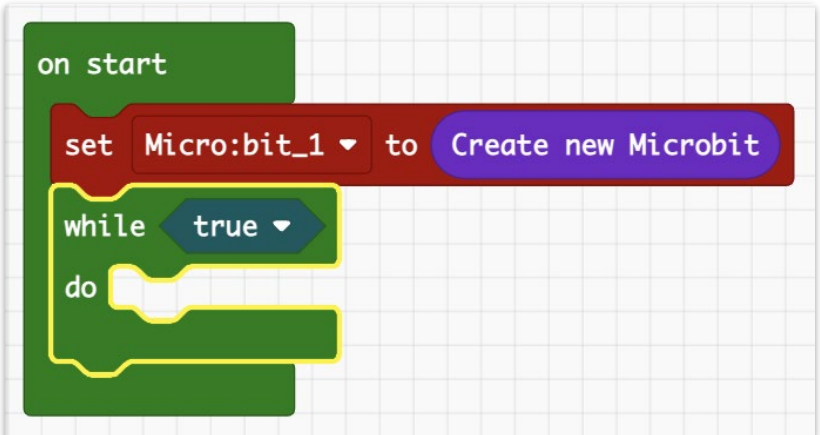






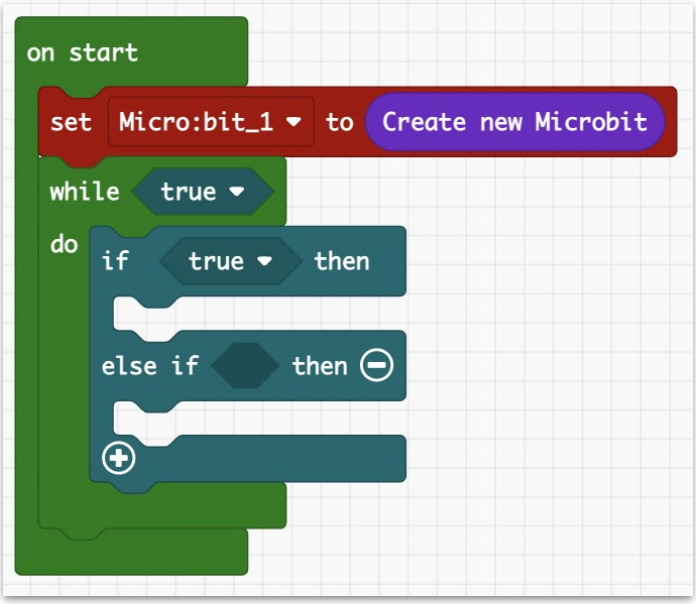


2. 從 “**Loops**”(迴圈)選項中， 拖曳一個“**while, do**” 迴圈作為下一步，將其設定為“**true**.”

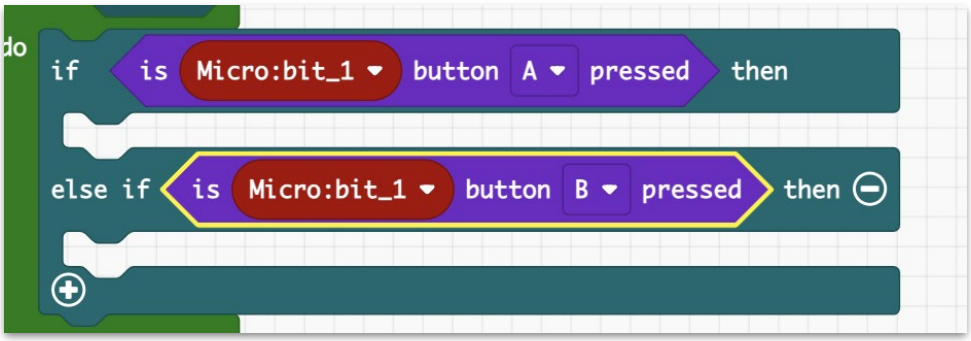


練習程式碼 3:
“奇怪知識驗證器”
(第一部分)

3. 從 “**Logic**”(邏輯)選項中，拖曳一個 “**if, do, else**”，並放入迴圈中。

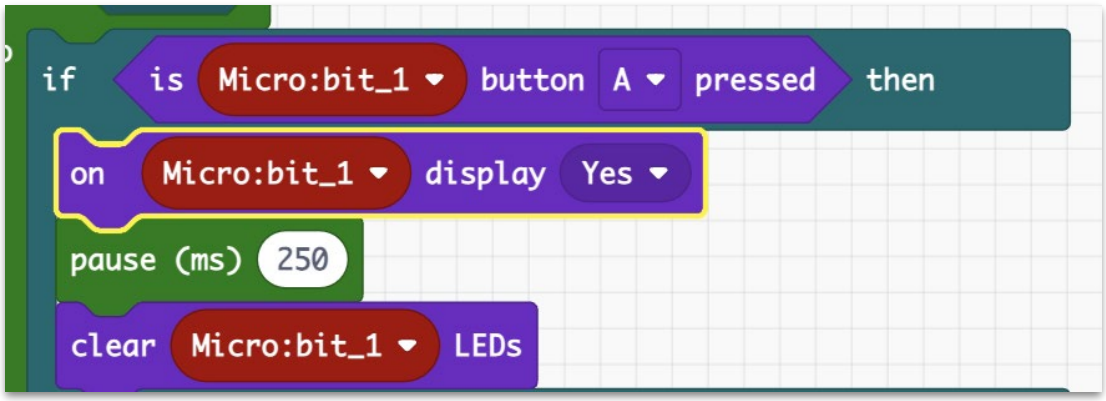


4. 從 “**Microbit**” 選項中，
   a. 拖曳一個“**is Microbit_1 button A pressed**” (Microbit_1 的按鈕A是否被按下)並增加到 “**if**” (如果)
   b. 拖曳一個 “**is Microbit_1 button A pressed**” (Microbit_1的按鈕A是否被按下並增加到 “**else if**” — 將 “A” 改成 “B.”。

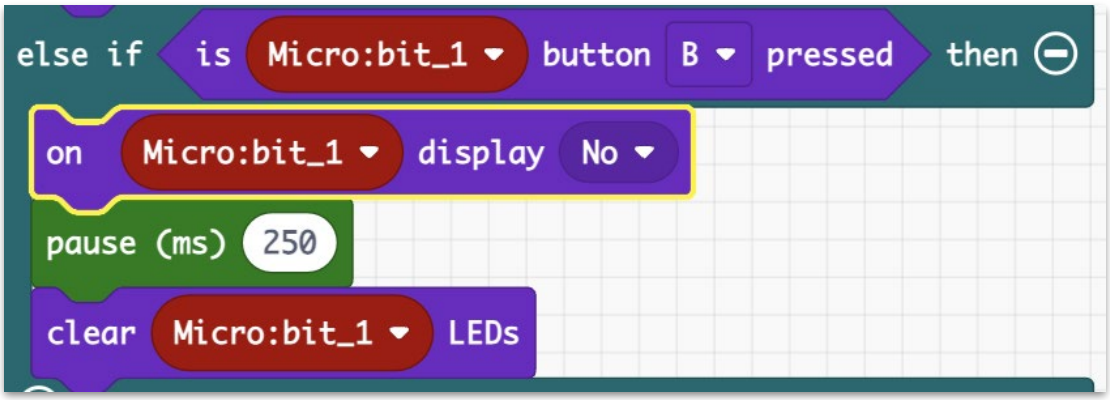


練習程式碼 3:
“奇怪知識驗證器”
(第一部分)

5. 從以下的選單中，新增：
   a. “Microbit” -拖曳一個“on Microbit_1 display “Yes” (在Microbit_1顯示” Yes”)，加到“if” (如果)中。
   b. “Loops”(迴圈)- 拖曳一個 “pause (ms) [250]”(暫停(ms)[250])。
   c. “Microbit” - 拖曳一個“clear Microbit_1 LEDs”(清除Microbit_1的LED燈)

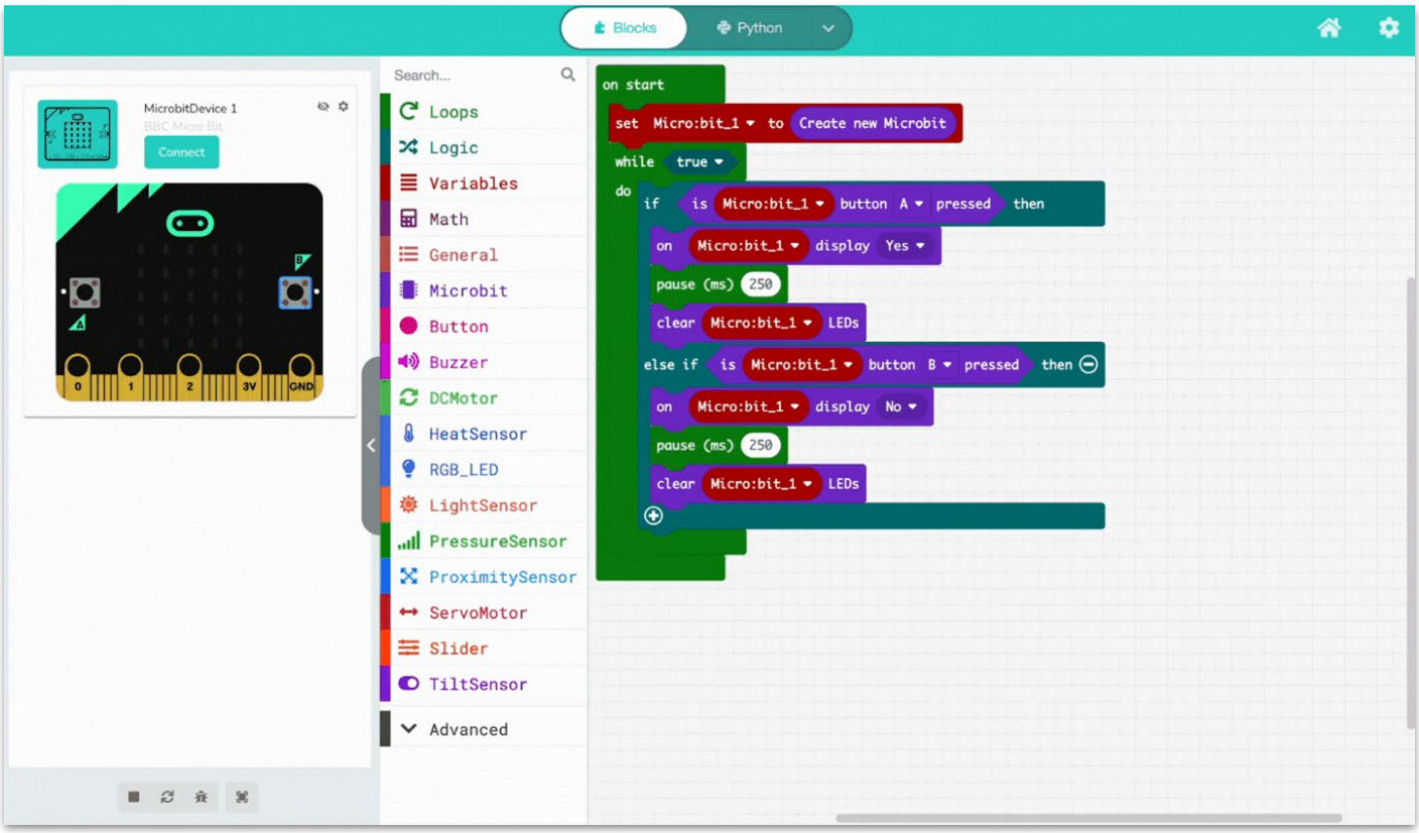


5. 從以下的選單中，新增：
   a. “Microbit” - 拖曳一個 “on Microbit_1 display “No” (在Microbit_1顯示” No “)，並加到 “else if” (否則)中。
   b. “Loops”(迴圈)- 拖曳一個 “pause (ms) [250]” (暫停(ms)[250])。
   c. “Microbit” - 拖曳一個 “clear Microbit_1 LEDs” (清除Microbit_1的LED燈)



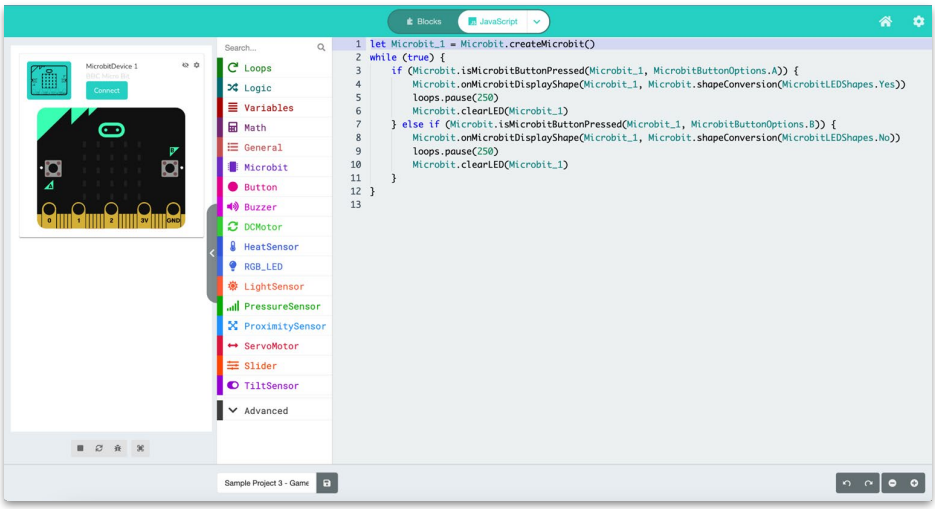
練習程式碼 3:
“奇怪知識驗證器”
(第一部分)

7. 測試程式。按下播放按鈕，查看程式。按下“A”鍵可以看到一個勾勾，按下“B”鍵可以看到一個“X:”。

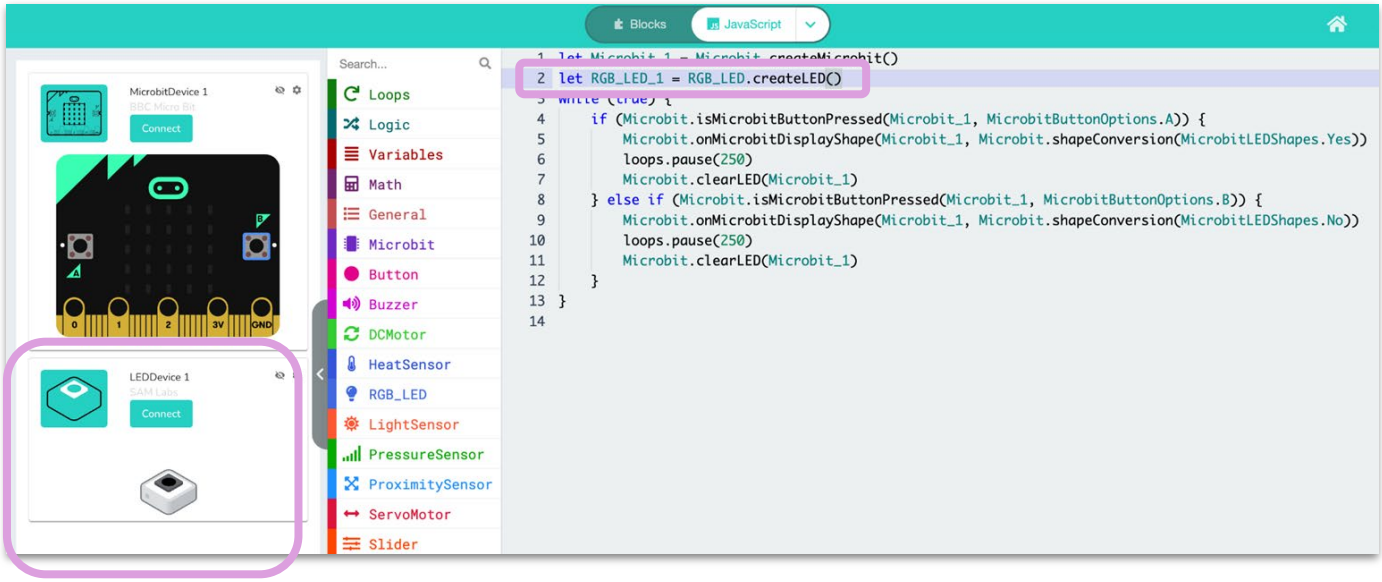


練習程式碼 3:
“奇怪知識驗證器”
(第二部分)

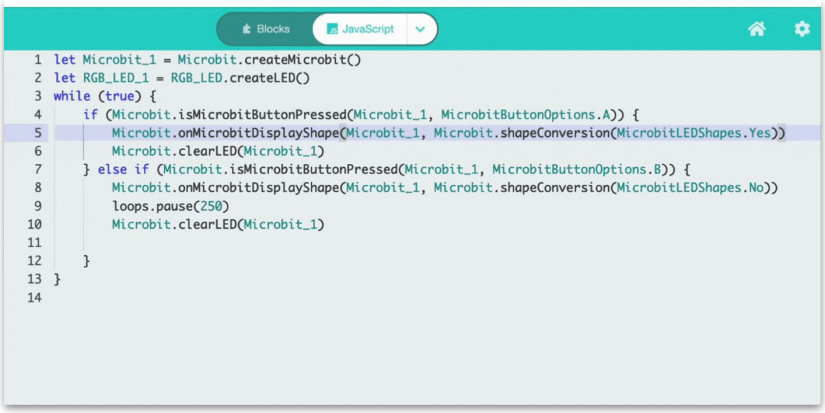
8. 選擇頂部選單中的 JavaScript 進行轉換：



9. 在程式碼中新增一個 RGB LED：
   a. 點擊第一行的末尾，開始新增第二行。
   b. 輸入：“`let RGB_LED_1 = RGB_LED.createLED()`”
   c. 測試您的程式碼；應該會出現一個 RGB LED。



練習程式碼 3:
“奇怪知識驗證器”
(第二部分)



10. 編寫程式碼，使 RGB LED 在按下Button A 閃爍為 Green (綠色)：
    a. 點擊第 5 行的末尾，在條件語句中開始新增第 6 行。
        - 輸入：“`RGB_LED.setLEDColor(RGB_LED_1, samLedColors.Green)`”
    b. 點擊第 8 行的末尾，在條件語句中開始新增第 9 行。
        - 輸入：“`RGB_LED.turnLEDOff(RGB_LED_1)`”
    c. 測試您的程式碼；當按下按鈕 A 時，燈應該閃爍為綠色。

11. 編寫程式碼，使 RGB LED 在按下Button B 時閃爍為Red(紅色)：
    a. 點擊第 10 行的末尾，在條件語句中開始新增第 11 行。
        - 輸入：“`RGB_LED.setLEDColor(RGB_LED_1, samLedColors.Red)`”
    b. 點擊第 13 行的末尾，在條件語句中開始新增第 14 行。
        - 輸入：“`RGB_LED.turnLEDOff(RGB_LED_1)`”
    c. 測試您的程式碼；當按下按鈕 B 時，燈應該閃爍為紅色。

練習程式碼 3:
“奇怪知識驗證器”
(第二部分)

12. 測試程式。按下播放按鈕，查看程式。按下「A」鍵可以看到一個勾勾，按下「B」鍵可以看到一個「X」。