

# 人人可編碼 幼兒教育篇

教師指南



## 目錄

### 簡介

### 指令

- 第 1 課：日常習慣
- 第 2 課：故事次序
- 第 3 課：舞步

### 函數

- 第 1 課：紙寶石
- 第 2 課：歌唱大會
- 第 3 課：我的平靜心靈函數

### 迴圈

- 第 1 課：重複製作花瓣
- 第 2 課：障礙賽道
- 第 3 課：鼓聲模式

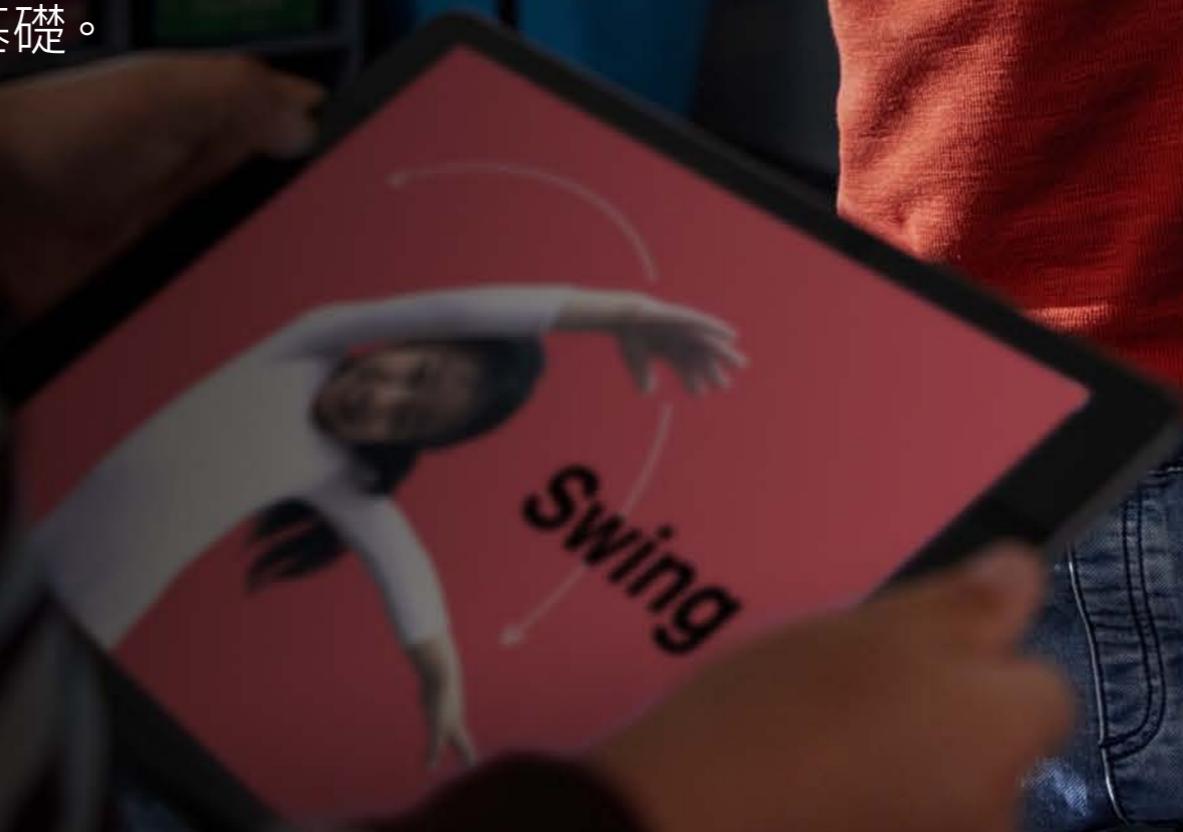
### 變數

- 第 1 課：浮或沉
- 第 2 課：文字遊戲
- 第 3 課：我的個人資料

### App 設計

### 導師資源

「人人可編碼：幼兒教育篇」旨在協助教育工作者和家庭，在學習者就讀低年級並首次發展各種運算思維技能時，為他們介紹編碼。透過這些課程，就讀幼稚園至三年級的學習者將以探索、發掘和遊戲的方式，建立核心編碼概念的基礎。



## 教學設計

這份指南分為四個單元，以及最終要完成的 app 設計專案。每個單元包含三節課，每節課都重點講解與編碼相關的一個概念。每節課內有三個活動：探索、發掘和遊戲。活動可以分拆為多節或分幾天完成。

### 第 1 日：討論與實踐學習

#### 探索

介紹並討論編碼概念

#### 發掘

透過創意活動熟悉概念

~25  
分鐘

#### 遊戲

- 在 Swift Playgrounds app 與 Byte 一起編碼
- 在隨附的工作紙和 Keynote 活動中練習編碼
- 透過離線的地面闖關編碼遊戲，將 Byte 世界帶入現實世界

~25  
分鐘

第 2 日：將學習和程式碼結合在一起

## 課程範圍和順序

這份指南中的四個單元適合幼稚園至三年級的學習者參與，可以按任意次序完成。隨著學習者加深對編碼和 app 的理解，建議你隨時或甚至一年內多次使用「App 設計」單元。

例子：

級別	單元	最終要完成的專案	大約總時間
幼稚園	指令	App 設計	4 小時
一年級	函數	App 設計	4 小時
二年級	迴圈	App 設計	4 小時
三年級	變數	App 設計	4 小時

## 繼續學習

至於四至八年級學生，「人人可編碼：解謎闖關」和「人人可編碼：探險闖關」課程連同《App 設計日誌》和《App 展示活動指南》共提供了 90 多個小時的學習機會。不妨在《[人人可編碼課程指南](#)》中了解更多。



**學習者作品集 (可選)**

從這些單元的活動中收集創作成果，與學習者一起建立作品集。



單元	課程	建議創作成果
指令	日常習慣	<ul style="list-style-type: none"> <li>• 「發出指令」工作紙</li> <li>• 「加入新指令」工作紙</li> </ul>
	故事次序	<ul style="list-style-type: none"> <li>• 「故事次序」情節點圖片</li> <li>• 「故事次序」小組圖片</li> </ul>
	舞步	<ul style="list-style-type: none"> <li>• 「舞步」卡</li> <li>• 「舞步」影片 (選用)</li> </ul>
函數	紙寶石	<ul style="list-style-type: none"> <li>• 紙寶石形狀</li> <li>• 「組合新的動作」工作紙</li> <li>• 「建立新函數」工作紙</li> </ul>
	歌唱大會	<ul style="list-style-type: none"> <li>• 歌唱大會音樂會影片或編寫函數</li> </ul>
	我的平靜心靈函數	<ul style="list-style-type: none"> <li>• 我的平靜心靈函數繪畫或影片</li> <li>• 「收集、切換、重複」工作紙</li> </ul>
迴圈	重複製作花瓣	<ul style="list-style-type: none"> <li>• 重複製作花瓣</li> <li>• 「使用迴圈」工作紙</li> <li>• 「迴圈每一側」工作紙</li> </ul>
	障礙賽道	<ul style="list-style-type: none"> <li>• 障礙賽道影片或圖片 (選用)</li> </ul>
	鼓圈	<ul style="list-style-type: none"> <li>• 「走到邊緣再返回」工作紙</li> <li>• 鼓圈影片或圖片 (選用)</li> </ul>
變數	浮或沉	<ul style="list-style-type: none"> <li>• 浮或沉</li> <li>• 「持續追蹤」工作紙</li> </ul>
	文字遊戲	<ul style="list-style-type: none"> <li>• 文字遊戲</li> </ul>
	我的個人資料	<ul style="list-style-type: none"> <li>• 我的個人資料</li> <li>• 你的個人資料</li> </ul>
App 設計		<ul style="list-style-type: none"> <li>• 什麼是 App?</li> <li>• 我的 App 設計</li> <li>• 「App 設計」原型</li> </ul>

## 在 iPad 或 Mac 開始使用 Swift Playgrounds



上課之前，請務必先下載 [Swift Playgrounds](#)、[Pages](#) 和 [Keynote](#)。

本指南的單元採用不同的 playground 組合。完成每個單元的所需條件如下：

單元	各種 Playground	如何在 Swift Playgrounds 中下載
指令	 學習程式設計 1  玩 MeeBot 學編舞	<p>要訂閱 MeeBot Playgrounds 資訊提供，請捲動到「More Playgrounds」畫面底部，然後點一下「輸入訂閱 URL」。然後輸入：<a href="https://ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json">ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json</a>。</p>
函數	 學習程式設計 1	
迴圈	 學習程式設計 1  玩 MeeBot 學編舞	<p>要訂閱 MeeBot Playgrounds 資訊提供，請捲動到「More Playgrounds」畫面底部，然後點一下「輸入訂閱 URL」。然後輸入：<a href="https://ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json">ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json</a>。</p>
變數	 學習程式設計 2  剪刀、石頭、布  程式碼機器	<p>你可以在「更多 Playground」畫面的「書籍」部分找到「剪刀、石頭、布」和「程式碼機器」。</p>
App 設計		

請在 [App Store](#) 中確認 Swift Playgrounds 的最低要求。瀏覽 [Apple 支援](#) 以取得 Swift Playgrounds 的說明。

## 導師貼士

為了讓學習者從課程中學到最多，不妨嘗試這些貼士。

### 「探索」和「發掘」活動：

- 在編寫或顯示程式碼時簡化任何語法或特殊大小寫，例如：
  - `var names = ["Rose", "Sam", "Joy"]` --> `var names = Rose, Sam, Joy`
  - `var ages = [7, 8, 7, 8, 7]` --> `var ages = 7, 8, 7, 8, 7`
  - `var myFavoriteColor = ■` --> `var my favorite color = ■`

### 「遊戲」活動：

- 為了讓幼兒學習者更容易掌握 Swift Playgrounds app，請遵循課程計劃裡的指示，包括：
  - 全組一起閱讀簡介
  - 為學習者提供隨附工作紙的精簡指示，以便他們構思自己的解決方案
  - 使用一部導師 iPad 或 Mac 以解決 app 裡的關卡
- `let` 和 `var`：這份指南沒有講解關鍵字 `let`。為免在 Swift Playgrounds 中出現混淆情況，請在向學習者展示頁面前將所有關鍵字 `let` 更改為 `var`。在我們推薦的 playground 中，這兩個關鍵字可以互換。
  - `let` = 變數不改變
  - `var` = 變數改變

### 延伸學習：

- 擴展地面「遊戲」活動，以加入計算、讀寫、視覺詞、拼寫等內容。在「函數」單元嘗試進行地面「遊戲」活動，以獲得靈感。
- 讓學習者製作自己的指令卡，例如 `twirl()` 或 `jump()`，來建立個人化的地面「遊戲」活動。



簡介頁面



Playground 頁面



# 指令



## 概覽

### 第 1 課：日常習慣

- 探索：在討論中將烘焙與指令聯想在一起
- 發掘：「日常習慣」活動
- 遊戲：「發出指令」和「加入新指令」

### 第 2 課：故事次序

- 探索：在討論中找出故事情節次序與指令之間的關係
- 發掘：「故事次序」活動
- 遊戲：地面闖關遊戲

### 第 3 課：舞步

- 探索：在討論中找出舞步與指令之間的關係
- 發掘：「舞步」活動
- 遊戲：「你好 MeeBot」和「基本舞步」

## 學習者將能夠

- 使用日常例子描述逐步指示
- 順序排列指示，使其變得合理
- 測試指示與程式碼並除錯

## 詞彙

- **Sequence (序列)**：事情發生的次序
- **Step (步驟)**：較大程序裡的一個動作
- **Modify (修改)**：即改變
- **Command (指令)**：告訴應用程式執行特定動作的程式碼
- **Bug (錯誤)**：程式碼中的錯誤
- **Debug (除錯)**：找出並修正程式碼中的錯誤

## 標準

1A-AP-08, 1A-AP-10, 1A-AP-12, 1A-AP-14, 1B-AP-16 >

### 探索

**目標：**將指令聯想成烘焙布朗尼蛋糕，介紹指令的概念。

#### 討論：

- 他們會參考布朗尼食譜嗎？
- 他們會順序完成食譜中的步驟嗎？

**要點：**食譜裡的每個步驟或指示就像程式碼的指令。讓學習者自行構思指令。

### 發掘

**目標：**確定逐步指示，建立日常習慣的流程模式。

**教材：**「洗手」卡

#### 指示：

1. 將一副「洗手」卡洗勻，然後放在桌上或白板上。卡片次序應已調亂。
2. 詢問學習者是否認為你的洗手次序有錯誤。
3. 請學習者將卡片移到正確位置，一次一張，對這些指示進行除錯 (即修正)。

#### 其他選擇：

請學習者以兩人一組或小組方式合作，並為每組派發一套卡片。

#### 延伸學習：

讓學習者為自己的某項日常活動構思一套逐步指示，並把每個步驟畫成圖片。

↓ [下載「洗手」卡](#)



## 遊戲

**目標：**學習者將能夠在 Swift Playgrounds app 的「學習程式設計 1」按正確次序加入指令，以收集自己的首顆寶石。

### 指示：

1. 將「學習程式設計 1」playground 中「指令」章節的簡介頁面投影到螢幕上。
2. 簡介：
  - 全班一起細閱各頁，必要時停下來解答疑問。
3. 發出指令：
  - 查看學習者指引 Byte 移往寶石的兩個所需指令，即 `moveForward()` 和 `collectGem()`。
  - 請學習者試驗各種方式，以指引 Byte 從開始箭嘴接近並收集寶石。他們可以在工作紙或另一張紙上記錄指令。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以完成關卡。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的構思。
  - 與 Byte 一起慶祝！

### 延伸學習：

如果學習者已準備就緒，就可以繼續前往下個頁面：「加入新指令」。在此處，學習者會使用新指令：`turnLeft()`。



學習程式設計 1

### 導師教材：

- iPad 或 Mac
- Swift Playgrounds app
- 「學習程式設計 1」playground
- 投影機或顯示器

### 學習者教材：

- 「發出指令」和「加入新指令」工作紙
- 鉛筆
- 額外紙張 (選用)

 [下載「學習編碼」工作紙](#)



### 探索

**目標：**探索書籍如何依照順序 (從頭到尾) 寫成，讓故事情節合情合理。

#### 討論：

- 詢問學習者書籍是否依照順序寫成。
- 如果書籍的開題、中間和結尾次序調亂，會發生什麼事？
- 探索幾個例子。

**重點：**找出跟程式碼的關係，強調將編碼指令按正確次序排列非常重要，就像故事的情節點一樣。



### 發掘

**目標：**學習者將故事中的各個情節點畫成圖片後，便可將圖片順序排列來正確重現故事。

#### 導師教材：

- 白板
- 箱頭筆

#### 學習者教材：

- 紙張
- 箱頭筆或顏色筆
- 其他選擇：iPad 裝置和繪畫 app

#### 指示：

1. 讀出一個學習者耳熟能詳的故事。全班一起決定故事的主要情節點。最好構思四至六個情節點。
2. 根據情節點數量，建立有相同學習者人數的小組，例如四個情節點就是每組四名學習者。
3. 讓組內每位學習者畫出其中一個情節點。
4. 各組輪流站到教室前，而學習者要拿著次序調亂的故事情節圖片。
5. 觀眾重新排列圖片，每次移動一張。
6. 學習者的圖片次序正確時，為每組拍下相片。

#### 延伸學習或其他選擇：

每組學習者處理不同的故事，畫圖前全組一起決定情節點。

## 遊戲

**目標：**學習者將能夠使用路線指令，以指引 Byte 穿過實體格子接近寶石。

**準備：**學習者會以三人一組形式合作。使用畫家膠紙，為每組在地上貼上四乘四的格子。

### 指示：

1. 分發教材，並將學習者分為三人一組。
2. 詳閱每個角色，然後為每位小組成員分配角色，準備玩第一個遊戲。
3. 讓學習者從設計師角色開始玩遊戲。
4. 玩三次遊戲，每次輪換角色卡。

### 角色：

- 設計師：將寶石與開始箭嘴放在格子上。
- 程式設計員：在同學幫助下，將指令卡放在格子的裡面或旁邊，引導 Byte 接近並收集寶石。
- 測試員：從箭嘴上的 Byte 開始，按照指令卡在格子內移動 Byte。如果收集到寶石，恭喜你！如果未能收集，請全隊一起除錯或修正編碼。

### 其他選擇：

如果學習者獨自與你進行活動或在家學習，則可以使用可下載的其他 Keynote 活動自行玩此遊戲。

### 導師教材：

- 畫家膠紙

### 學習者教材：

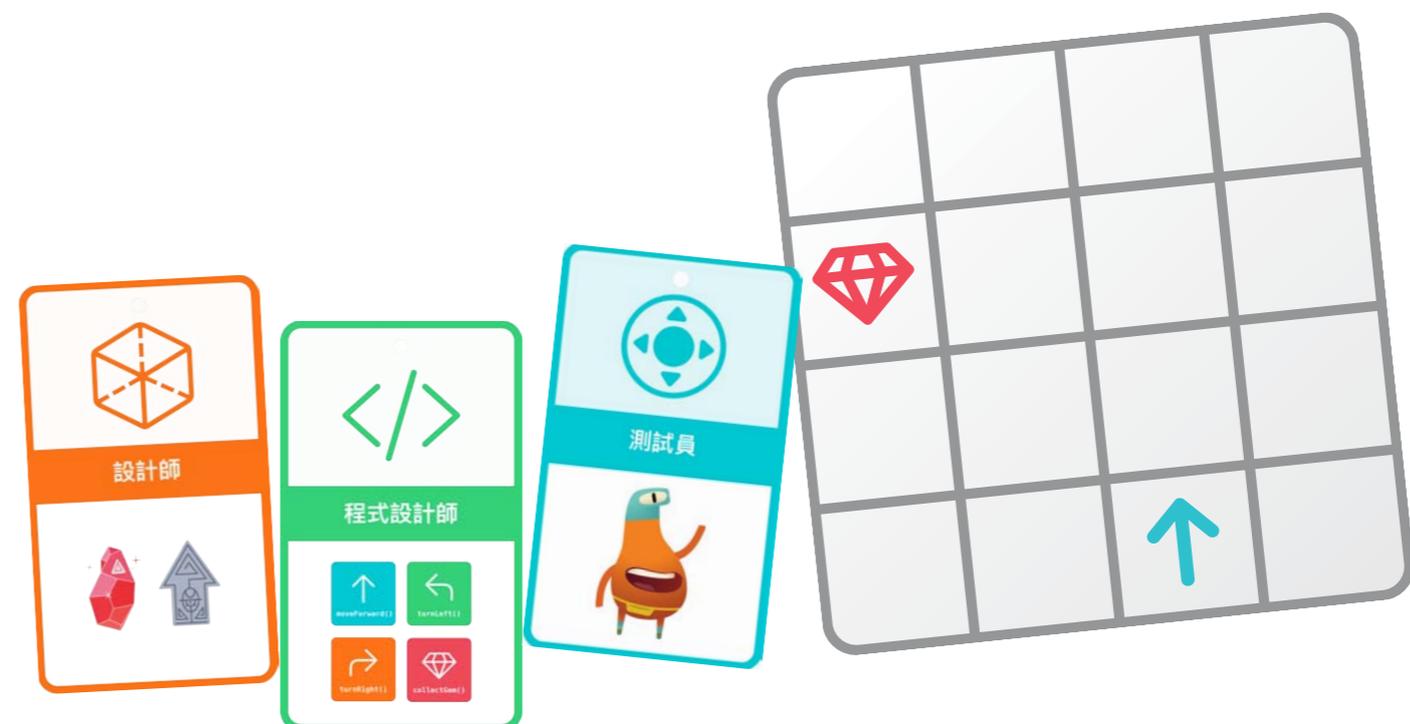
- 角色卡
- 指令卡：`moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()`
- 寶石
- Byte
- 箭嘴



下載教材



下載其他活動選擇



### 探索

**目標：**探索如何在編碼時發揮創意！

#### 討論：

- 詢問學習者是否學過跳舞。
- 舞蹈是否有順序步驟要跟從？
- 他們知道下一步該怎麼做嗎？
- 舞步有名稱嗎？
- 學習者有沒有試過，在同一舞蹈的不同時間或在不同舞步組合中使用相同舞步？

**要點：**幫助學習者相信他們可以在編碼時發揮創意，而且編碼人員可以製作新指令並將指令以各種有趣的方式組合起來，就像編排舞蹈一樣。

### 發掘

**目標：**創作簡短的舞步組合，以及建立展示舞步的卡片。每張「舞步」卡片就像「學習編碼」playground 中的指令。

#### 學習者教材：

- iPad 裝置
- Keynote app
- 相機 app
- 跳舞空間

#### 指示：

1. 讓學習者以兩人一組或小組形式創作簡短的舞步組合。
2. 學習者建立組合後，便可將不同舞步製作成卡片。學習者需在每張卡片上畫出並命名舞步，同時盡展創意構思趣怪的舞蹈動作。
3. 每組表演自己的舞蹈，全班一起來個舞蹈派對！

#### 其他選擇：

學習者可以使用下方的可下載「舞步」卡來自行編舞，或者參考這些卡片來自製卡片。

#### 延伸學習：

學習者製作自己的跳舞影片，向小組成員展示。

 [下載「舞步」卡](#)



## 遊戲

**目標：**建立一系列步驟，以教授 MeeBot 機械人跳新舞。

### 指示：

1. 將「玩 MeeBot 學編舞」playground 投影到螢幕上。如果還未訂閱此 playground，請先訂閱。
2. 簡介：
  - 全班一起細閱各頁，必要時停下來解答疑問。
3. 你好 MeeBot：
  - 按一下或點一下「執行我的程式碼」，觀看機械人跳舞。
4. 基本舞步：
  - 學習者以小組、兩人一組或自行使用 iPad 獨自工作的形式，從建議列表中選擇八個指令，然後觀看機械人跳舞。
  - 請學習者分享自己的舞蹈，或全班一起編排數種不同舞蹈。
  - 與機械人一起跳舞！

### 延伸學習：

- 繼續前往下個頁面「舞步組合」，學習者可以在 `myDanceRoutine()` 函數內新增舞步，隨心加入任何數量的指令。



玩 MeeBot 學編舞

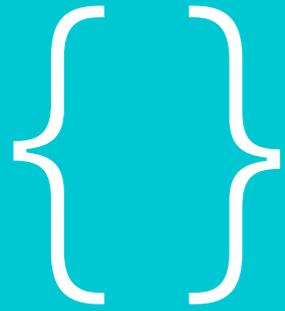
### 導師教材：

- iPad 或 Mac
- Swift Playgrounds app
- 「玩 MeeBot 學編舞」playground
- 投影機或顯示器

### 學習者教材：

- iPad 裝置 (選用)





# 函數



## 概覽

### 第 1 課：紙寶石

- 探索：討論逐步指示
- 發掘：「紙寶石」活動
- 遊戲：「組合新的動作」和「建立新函數」

### 第 2 課：歌唱大會

- 探索：討論如何為函數命名
- 發掘：「歌唱大會」活動
- 遊戲：地面闖關遊戲

### 第 3 課：我的平靜心靈函數

- 探索：討論如何以多種方式解決問題
- 發掘：「我的平靜心靈函數」活動
- 遊戲：收集、切換、重複

## 學習者將能夠

- 將大型問題或任務解構為小型步驟
- 建立一系列步驟以解決問題或完成任務
- 為函數命名
- 測試程式碼並除錯

## 詞彙

- **Function (函數)**：可在需要時執行的一組已命名指令
- **Toggle (切換)**：開啟或關閉

## 標準

1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14, 1B-AP-16 >

## 探索

**目標：**探索將一系列指令組合起來並命名的構思。

**討論：**全班一起決定要集中討論的日常習慣。讓學習者確定其日常習慣的名稱，以及完成日常習慣的步驟。

**例子：**就寢習慣

- 第 1 步：刷牙
- 第 2 步：使用洗手間
- 第 3 步：閱讀
- 第 4 步：說晚安
- 第 5 步：關燈

**要點：**構思並命名一組指示，與建立函數的概念相同。

**延伸學習：**詢問學習者，他們是否有任何步驟的指示可以更加具體。例如，刷牙的詳細步驟是什麼？

## 發掘

**目標：**學習者首先按照指示製作紙寶石，然後為自己選擇的另一個形狀寫出或繪畫指示。

**學習者教材：**

- 紙張
- 剪刀
- 鉛筆
- iPad 裝置 (選用)

**指示：**

向學習者示範如何製作紙寶石：

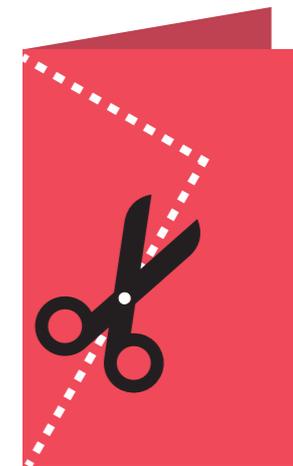
1. 將一張紙對摺。
2. 從摺起一邊的頂角畫一條線，去到紙張中心上方一至兩吋。
3. 從第一條線末端畫另一條線，去到摺起一邊的底角。
4. 沿著畫出的線剪下來。
5. 將寶石從紙上取下並展開。

請學習者製作自己的形狀：

1. 將學習者分成小組。
2. 讓小組決定要製作什麼形狀。
3. 預留時間讓學習者練習製作形狀一至兩次。
4. 學習者寫出或繪畫製作形狀的指示並為其命名，例如「製作圓圈」或「字母 T」。

**其他選擇：**

製作影片展示如何製作形狀。



## 遊戲

**目標：**全體學習者一起合作，拆分讓 Byte 移往寶石所需的步驟。

**指示：**

1. 將「學習程式設計 1」playground 投影到螢幕上。前往「學習程式設計 1」中的「函數」章節。
2. 簡介：
  - 全班一起細閱各頁，必要時停下來解答疑問。
3. 組合新的動作：
  - 查看指令 `moveForward()`、`turnLeft()` 和 `collectGem()` — 請記住，你沒有 `turnRight()` 指令。
  - 讓學習者試驗各種方式，以指引 Byte 從開始箭嘴接近並收集寶石。他們可在工作紙或另一張紙上記錄指令。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以完成關卡。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的構思。
  - 與 Byte 一起慶祝！
4. 建立新函數：
  - 根據在上個 Playground 頁面學到的知識來組合新動作，讓學習者構思各種方法來建立 `turnRight()` 函數。
  - 使用 `turnRight()` 函數，讓學習者試驗各種方式，以指引 Byte 從開始箭嘴接近並切換關閉開關。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以完成關卡。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的構思。
  - 與 Byte 一起慶祝：你剛過了高難度關卡！



學習程式設計 1

**導師教材：**

- iPad 或 Mac
- Swift Playgrounds app
- 「學習程式設計 1」playground
- 投影機或顯示器

**學習者教材：**

- 「組合新的動作」和「建立新函數」工作紙
- 鉛筆
- 額外紙張 (選用)



[下載「學習編碼」工作紙](#)

### 探索

**目標：**為歌曲取個一看就懂的名稱，把指令和函數知識應用到歌曲中。

**討論：**讓學習者想出各種歌曲，並為每首歌取個一看就懂的函數名稱。

**例子：**對於歌曲「Twinkle, Twinkle, Little Star」函數呼叫可以是 `singTwinkle()`，但 `singSong1()` 就算不上好名稱，理由是第一首歌曲或會有變。

**要點：**為函數取個一看就懂的名稱很重要，因為這能讓你和其他人更容易理解程式碼。

### 發掘

**目標：**學習者呼叫音樂會函數裡的不同歌曲指令，來建立音樂會。

**導師教材：**

- iPad 或 Mac
- 投影機或顯示器
- 白板
- 箱頭筆

**指示：**

1. 幫助學習者為幾首歌曲建立函數名稱，例如 `singHappyBirthday()`。
2. 全組一起選擇唱歌的次序。
3. 為音樂會編寫函數定義，並用歌曲指令填寫函數。

**例子：**

```
func createConcert() {
    singHappyBirthday()
    singTwinkleTwinkle()
    singMaryHadALittleLamb()
}
createConcert()
```

**其他選擇：**

學習者以小組形式唱歌，每組構思自己的歌單、歌曲函數名稱以及唱歌次序。然後，每組表演自己的歌曲並製作演唱會的影片。

## 遊戲

**目標：**學習者將解出簡單的方程式，把寶石放在答案上，然後使用路線指令指引 Byte 在格子上移動。

**準備：**學習者會以三人一組形式合作。使用畫家膠紙，為每組在地上貼上四乘四的格子。將開始箭嘴放在一個正方格內，然後在剩下的每個正方格內放置一個數字。

### 指示：

1. 分發教材，並將學習者分為三人一組。
2. 詳閱每個角色，然後為每位小組成員分配角色，準備玩第一個遊戲。
3. 讓學習者從設計師角色開始玩遊戲。
4. 玩三次遊戲，每次輪換角色卡。

### 角色：

- 設計師：擲兩粒骰。在同學幫助下，將兩個數字相加，並將寶石放在含有總數的格子方格上。
- 程式設計員：在同學幫助下，將指令卡放在格子的裡面或旁邊，引導 Byte 接近並收集寶石。
- 測試員：從箭嘴上的 Byte 開始，按照指令卡在格子內移動 Byte。如果收集到寶石，恭喜你！如果未能收集，請全隊一起修正編碼。

### 其他選擇：

如果學習者獨自與你進行活動或在家學習，則可以使用可下載的其他 Keynote 活動自行玩此遊戲。

### 導師教材：

- 畫家膠紙
- 每個格子一組打印的數字

### 學習者教材：

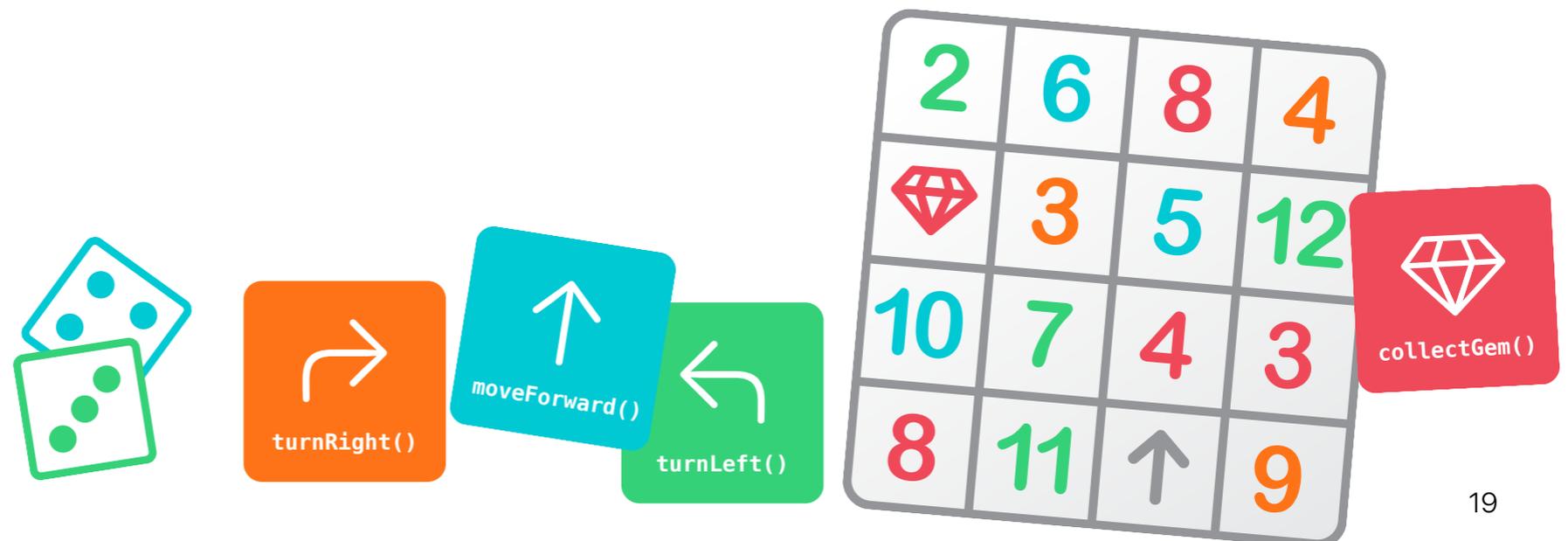
- 角色卡
- 指令卡：`moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()`
- 寶石
- Byte
- 箭嘴
- 兩粒骰



下載教材



下載其他活動選擇



### 探索

**目標：**學習者將能理解，解決問題的方法通常不止一個。

**討論：**讓學習者思考遇過的問題，然後分享解決辦法。詢問小組是否有人可以用不同的方式解決該問題。探索幾種問題與解決方案。

**要點：**幫助學習者與程式碼建立關係，並明白解決程式設計問題的方法通常不止一個。

### 發掘

**目標：**學習者會為自己平靜心靈的技巧編寫函數並為其命名。

**學習者教材：**

- 「我的平靜心靈函數」工作紙
- 鉛筆
- 彩色畫筆或顏色筆

**指示：**

**貼士：**如果可以，學習者最好獨立完成此活動。

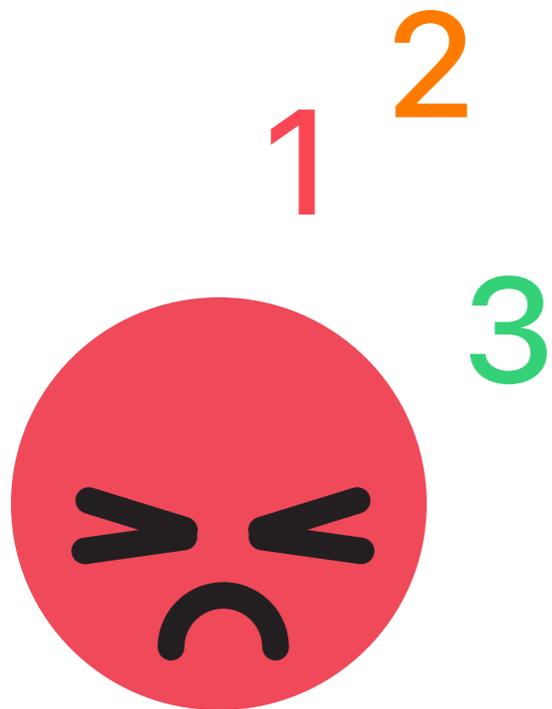
1. 請學習者集思廣益，構思他們在家或在學校心煩意亂時讓自己冷靜下來的辦法。學習者將自己平靜心靈的技巧分拆成各個步驟。
2. 分發「我的平靜心靈函數」工作紙，並請學習者畫出自己平靜心靈技巧的步驟。
3. 請學習者為自己平靜心靈的技巧命名。他們可以使用駝峰式大小寫，例如 `countToTen()`，或者使用短句，例如「數到十。」

**延伸學習：**

**真實呈現：**請學習者以小組形式在全班面前示範他們平靜心靈的技巧。

**使用 iPad：**學習者製作自己平靜心靈技巧的影片，與全班分享。

↓ [下載「我的平靜心靈函數」工作紙](#)



## 遊戲

**目標：**學習者將能夠編寫由幾個不同指令類型組成的函數，然後使用此函數完成關卡。

**指示：**

1. 將「學習程式設計 1」playground 中的「收集、切換、重複」頁面投影到螢幕上，指出學習者要幫助填寫的空白函數。
2. 收集、切換、重複：
  - 查看指令 `moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `toggleSwitch()`。
  - 讓學習者嘗試識別關卡中重複的部分，然後使用自己的構思去完成 app 中的函數並為其命名。
  - 讓學習者為函數發明符號，並在工作紙的指令鍵中記錄符號和函數名稱。
  - 透過附加指令，學習者可以試驗各種方式來指引 Byte 收集所有寶石並切換所有開關。他們可在工作紙或另一張紙上記錄指令。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以完成關卡。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的解決方案。
  - 全班一起慶祝：大家剛過了高難度關卡！



學習程式設計 1

**導師教材：**

- iPad 或 Mac
- Swift Playgrounds app
- 「學習程式設計 1」playground
- 投影機或顯示器

**學習者教材：**

- 「收集、切換、重複」工作紙
- 鉛筆
- 額外紙張 (選用)

↓ [下載「學習編碼」工作紙](#)





# 迴圈



## 概覽

### 第 1 課：重複製作花瓣

- 探索：在討論中將編碼的重複步驟與現實生活聯想在一起
- 發掘：「重複製作花瓣」活動
- 遊戲：「使用迴圈」與「迴圈每一側」

### 第 2 課：障礙賽道

- 探索：討論迴圈停靠點
- 發掘：「障礙賽道」活動
- 遊戲：地面闖關遊戲

### 第 3 課：鼓聲模式

- 探索：關於音樂迴圈的討論
- 發掘：「鼓聲模式」活動
- 遊戲：「走到邊緣再返回」與「舞蹈循環」

## 學習者將能夠

- 識別編碼裡的迴圈
- 將大型問題或任務解構為小型步驟
- 建立一系列指令並使用迴圈重複該序列
- 測試指示與程式碼並除錯

## 詞彙

- Loop (迴圈)：重複特定次數的一段程式碼

## 標準

1A-CS-01, 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14 >

## 探索

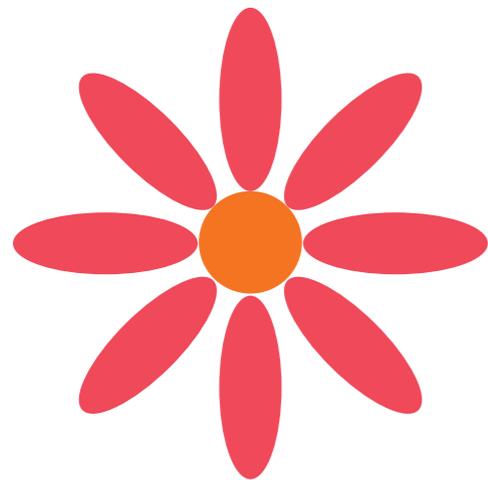
**目標：**將迴圈的構思與現實生活聯想在一起。

**討論：**探索學習者在現實生活中可能需要反覆完成某件事或某個步驟的情況。

**例子：**

- 步行
- 騎單車
- 縫紉、針織或鈎織

**要點：**迴圈可根據你指定的次數重複一個指令或一組指令。



## 發掘

**目標：**學習者製作獨一無二的花朵，從而開始探索迴圈的概念。

**學習者教材：**

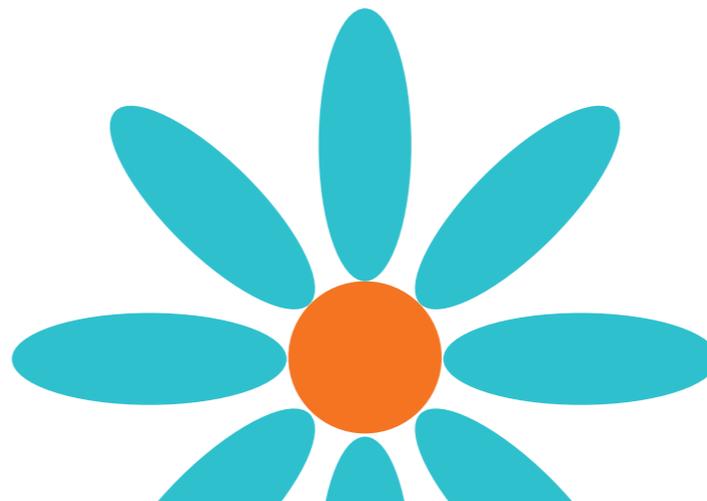
- 「重複製作花瓣」工作紙
- 色紙
- 鉛筆
- 剪刀
- 漿糊
- 骰子

**指示：**

1. 學習者在色紙畫出一片花瓣，花瓣長度大概等於自己手掌的長度，並將花瓣剪下來。這將會是他們花朵的花瓣樣板。
2. 每位學習者擲兩粒骰，將數字相加，然後在自己的「重複製作花瓣」工作紙填上迴圈遺漏的數字。這個數字就是他們花朵的花瓣數量。
3. 學習者使用花瓣樣板，在色紙上描繪花瓣，並為自己的花朵裁剪正確數量的花瓣。
4. 學習者使用「重複製作花瓣」工作紙，拼起花朵並將各個部分黏好。



[下載「重複製作花瓣」工作紙](#)



## 遊戲

**目標：**學習者將能夠編寫迴圈內的程式碼，以收集所有寶石。

### 指示：

1. 將「學習程式設計 1」playground 中「For 迴圈」章節的簡介頁面投影到螢幕上。
2. 簡介：
  - 全班一起細閱各頁，必要時停下來解答疑問。
3. 使用迴圈：
  - 向學習者展示傳送門如何運作，並查看指令 `moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()`。
  - 讓學習者試驗各種方式，以指引 Byte 從開始箭嘴接近並收集寶石，同時注意重複了哪些指令。他們可在工作紙或另一張紙上記錄指令。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以指引 Byte 收集首顆寶石並走向傳送門。
  - 詢問學習者有多少顆寶石，並在迴圈加上該數字。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的解決方案。
  - 與 Byte 一起慶祝！
4. 迴圈每一側：
  - 讓學習者試驗各種方式以收集所有寶石，同時注意重複用了哪些指令。
  - 使用編輯器底部的程式碼建議或點一下螢幕頂部的 + 以新增 `for` 迴圈。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以完成關卡。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的構思。
  - 與 Byte 一起慶祝！



學習程式設計 1

### 導師教材：

- iPad 或 Mac
- Swift Playgrounds app
- 「學習程式設計 1」playground
- 投影機或顯示器

### 學習者教材：

- 「使用迴圈」與「迴圈每一側」工作紙
- 鉛筆
- 額外紙張 (選用)



[下載「學習編碼」工作紙](#)

### 探索

**目標：**探索迴圈一律需要特定終點的原因。

**討論：**請學習者想像摩天輪或其他自己熟悉的遊樂設施。如果操作員五輪之後沒有按下按鈕讓設施停下來，會發生什麼事？讓學習者構思其他例子，說明如果迴圈不停止會發生什麼事。

**要點：**幫助學習者理解如果他們不停止迴圈，迴圈就會無限重複。

### 發掘

**目標：**學習者將循環執行自己設計的障礙賽道，以探索迴圈的運作原理。

**教材：**

- 進行體能活動的所需空間
- 障礙賽道道具
- 骰

**指示：**

1. 在教室或室外建立短程障礙賽道。
2. 擲一粒骰，然後讓學習者根據所擲數字重複跑完賽道。

**其他選擇：**

學習者構思出一系列動作，例如觸碰腳趾、跳躍、單腳踢。擲一粒骰，然後學習者根據所擲數字重複做一系列動作。



## 遊戲

**目標：**學習者將可建立含重複形式的關卡，然後全組一起破解。

**準備：**學習者會以三人一組形式合作。使用畫家膠紙，為每組在地上貼上四乘四的格子。

### 指示：

1. 分發教材，並將學習者分為三人一組。
2. 詳閱每個角色，然後為每位小組成員分配角色，準備玩第一個遊戲。
3. 讓學習者從設計師角色開始玩遊戲。
4. 玩三次遊戲，每次輪換角色卡。

### 角色：

- 設計師：在同學幫助下，將三顆寶石以重複形式置於格子上。將開始箭嘴放在格子上。
- 程式設計員：在同學幫助下，將指令卡放在格子的裡面或旁邊，引導 Byte 接近並收集寶石。使用「迴圈」卡告訴測試員指令循環執行的次數。
- 測試員：從箭嘴上的 Byte 開始，按照指令卡在格子內移動 Byte。如果收集到所有寶石，恭喜你！如果未能收集，請全隊一起修正編碼。

### 其他選擇：

如果學習者獨自與你進行活動或在家學習，則可以使用可下載的其他 Keynote 活動自行玩此遊戲。

### 導師教材：

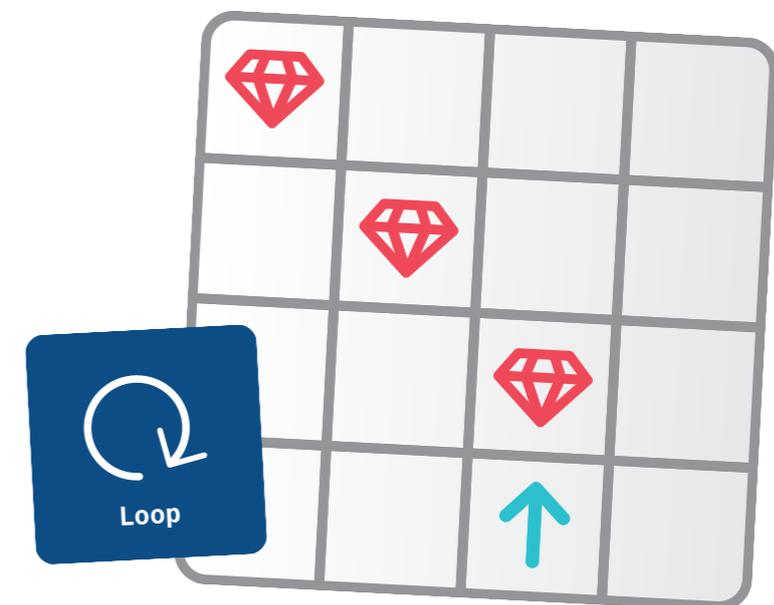
- 畫家膠紙

### 學習者教材：

- 角色卡
- 指令卡：`moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `Loop`
- 寶石
- Byte
- 箭嘴

↓ [下載教材](#)

↓ [下載其他活動選擇](#)



### 探索

**目標：**探索音樂中的重複模式。

**討論：**讓學習者分享他們演奏的樂器或唱過的歌曲。詢問學習者在演奏或唱歌時是否重複過節拍或副歌。他們是否可以想出歌曲或音樂中的其他重複部分？

**要點：**強調迴圈是由兩個部分組成：

- 指令
- 重複的次數

### 發掘

**目標：**學習者將可重複鼓聲模式，找出循環執行程式碼與現實生活實例之間的關係。

**教材：**

- 某些可以敲擊的物件，如地板、大腿或書籍
- 可以圍圈而坐的空間

**指示：**

1. 讓學習者圍坐成一個圓圈。
2. 讓學習者重複你創作的鼓聲節拍，重複次數與你舉起的手指數目相同。例如，如果你舉起四隻手指，學習者就應該重複鼓聲節拍四次然後停止。
3. 輪流讓圓圈內的學習者重複鼓聲節拍或分成小組，使每個人都有機會成為帶頭鼓手。

**延伸學習：**

讓學習者製作鼓聲。



## 遊戲

**目標：**學習者會呼叫迴圈內幾個不同的指令，並決定呼叫迴圈的次數。

**指示：**

1. 將「學習程式設計 1」playground 投影到螢幕上。前往「For 迴圈」章節第三頁的「走到邊緣再返回」。
2. 走到邊緣再返回：
  - 查看指令 `moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `toggleSwitch()`。
  - 讓學習者試驗各種方式，以指引 Byte 從開始箭嘴走向每個已關閉的開關，並打開開關。
  - 使用編輯器底部的程式碼建議或點一下螢幕頂部的 + 以新增 `for` 迴圈。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以完成關卡。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的構思。
  - 與 Byte 一起慶祝！
3. 離開「學習程式設計 1」並開啟「玩 MeeBot 學編舞」playground，前往「舞蹈循環」頁面。(此 playground 頁面沒有學習者工作紙。)
4. 舞蹈循環：
  - 讓學習者以小組、兩人一組或自行使用 iPad 獨自工作的形式，來完成迴圈並觀看機械人跳舞。
  - 請學習者分享自己的舞蹈，或全班一起建立幾種不同舞蹈。
  - 與機械人一起跳舞！



學習程式設計 1



玩 MeeBot 學編舞

**導師教材：**

- iPad 或 Mac
- Swift Playgrounds app
- 「學習程式設計 1」playground
- 「玩 MeeBot 學編舞」playground
- 投影機或顯示器

**學習者教材：**

- 「走到邊緣再返回」工作紙
- 鉛筆
- iPad 裝置 (選用)
- 額外紙張 (選用)



[下載「學習編碼」工作紙](#)



# 變數



## 概覽

### 第 1 課：浮或沉

- 探索：關於更新變數的討論
- 發掘：「浮或沉」活動
- 遊戲：「持續追蹤」和「遊戲範例」

### 第 2 課：文字遊戲

- 探索：關於問題答案類型的討論
- 發掘：「文字遊戲」活動
- 遊戲：地面闖關遊戲

### 第 3 課：我的個人資料

- 探索：關於使用列表回答問題的討論
- 發掘：「我的個人資料」活動
- 遊戲：「使用迴圈」

## 學習者將能夠

- 把變數名稱與既定值連結起來
- 更改分配給變數的值
- 理解可以分配給變數的不同 Swift 類型，包括 true/false (布林值)、number (整數)、word (字串)、color (顏色常值) 和 image (圖片常值)
- 測試指示與程式碼並除錯

## 詞彙

- **Variable (變數)**：具名容器，用於存放可變值
- **Data (數據)**：資料
- **Boolean (布林值)**：值為 true 或 false 的類型

## 標準

1A-AP-09, 1B-AP-09, 1B-AP-10, 1B-AP-16 >

## 探索

**目標：**透過點算物件數量與更新變數數字，探索變數的概念。

### 導師教材：

- 白板
- 箱頭筆
- 白板刷
- 容器
- 五支鉛筆 (或任何五件相同物件)

### 指示：

1. 首先，在白板上寫下變數語句以追蹤物件。
  - 例子：`var numberOfPencils = 0`
2. 拿起空置容器並告訴學習者此容器代表你的變數：`numberOfPencils`。
3. 放一支鉛筆到容器中，並詢問學習者現在變數數量是多少。若他們回答正確，擦掉 0 並寫上 1。
4. 繼續放入鉛筆，直至所有鉛筆都已放進容器，且程式碼為：`var numberOfPencils = 5`。
5. 然後，開始將鉛筆從容器內取出，並在拿走鉛筆時更新變數。

**要點：**幫助學習者理解變數儲存了一些資訊。在這個情況，資訊是一個數字，用來指出容器內有多少支鉛筆。

## 發掘

**目標：**學習者將使用找到的物件進行實驗，以決定物件會沉還是浮，然後使用 `image` (圖像常值) 和 `true/false` 值 (布林值) 記錄數據。

### 學習者教材：

- iPad 裝置
- Keynote app
- 「浮或沉」工作紙
- 一桶水
- 幾個測試物件

### 指示：

1. 將學習者分成小組。
2. 學習者收集不同的物件以供測試。
3. 對於每個物件，請學習者：
  - 為物件拍照並將其加至工作紙中。
  - 在水中測試物件。
  - 圈出 `true` 或 `false`，以在工作紙上記錄結果。



[下載「浮或沉」工作紙](#)

## 遊戲

**目標：**學習者將能夠在兩種編碼情境中建立並更新變數。

**指示：**

1. 將「學習程式設計 2」playground 投影到螢幕上。前往「變數」章節。
2. 簡介：
  - 全班一起細閱各頁，必要時停下來解答疑問。
3. 持續追蹤：
  - 讓學習者試驗各種方式，以指引 Hopper 從開始箭嘴接近並收集寶石。他們可在工作紙或另一張紙上記錄指令。
  - 請全班動動腦筋，在 Swift Playgrounds app 編寫程式碼以完成關卡。按一下或點一下「執行我的程式碼」。
  - 嘗試幾種不同的構思。
  - 與 Hopper 一起慶祝！
4. 離開「學習程式設計 2」，繼續前往「剪刀、石頭、布」playground 最後一頁的「遊戲範例」部分。（此 playground 頁面沒有學習者工作紙。）
5. 遊戲範例：
  - 在作出任何改變之前，先按一下或點一下「執行我的程式碼」玩遊戲。
  - 以小組形式討論你想自訂遊戲的哪些部分。你可以改變的一些有趣事情包括 `game.roundsToWin`、`game.challenger.emoji`、`game.addOpponent`，以及 `game.roundPrize`。
  - 多玩幾次遊戲，每次作出一些改變。

**延伸學習：**許多變數都建立在 Game.swift 檔案中。如果學習者好奇為什麼有些變數前面沒有 `var`，你可以開啟 Game.swift 檔案來指出遊戲屬性建立位置。



學習程式設計 2



剪刀、石頭、布

**導師教材：**

- iPad 或 Mac
- Swift Playgrounds app
- 「學習程式設計 2」playground
- 「剪刀、石頭、布」playground
- 投影機或顯示器

**學習者教材：**

- 「持續追蹤」工作紙
- 鉛筆
- 額外紙張（選用）



[下載「學習編碼」工作紙](#)

## 探索

**目標：**探索現實世界的多種答案類型，並將它們與不同的 Swift 類型聯想在一起，包括 yes/no 或 true/false (布林值)、number (整數)、word (字串)、color (顏色常值)，以及 image (圖片常值)。

### 導師教材：

- 白板
- 箱頭筆

**討論：**全班一起構思有多種答案的問題，並寫在白板上。

### 例子：

- 你的眼睛是什麼顏色？—> color
- 你是否有寵物？—> yes/no
- 你是否有兄弟姐妹？—> yes/no
- 你今年幾歲？—> number
- 你叫什麼名字？—> word

**要點：**解釋變數同樣有不同類型，包括數字、文字、顏色、圖片，以及「是或否」答案。視乎建立變數的方式，即使你更新變數內容，其類型亦要保持不變。例如，`var myAge = 8` 可以改為 9，但是無法改為「nine」（九）。

## 發掘

**目標：**學習者填入正確的答案類型，完成文字遊戲。

### 學習者教材：

- 「文字遊戲」工作紙
- 鉛筆
- 顏色筆

### 指示：

讓學習者以小組形式完成一個或多個文字遊戲。每組最好至少有一人懂得閱讀或充當支援。如果所有學習者都不擅閱讀，你可讓整組一起完成遊戲。

**延伸學習：**如果學習者有能力，請讓他們創作文字遊戲給拍檔填寫。鼓勵他們使用數字、文字、顏色、圖像和「是或否」答案來填充空位。



[下載「文字遊戲」工作紙](#)

## 遊戲

**目標：**學習者將能夠指引 Byte 收集多顆寶石，把每顆寶石加進容器並更新變數。

**準備：**學習者會以三人一組形式合作。使用畫家膠紙，為每組在地上貼上四乘四的格子。

### 指示：

1. 分發教材，並將學習者分為三人一組。
2. 詳閱每個角色，然後為每位小組成員分配角色，準備玩第一個遊戲。
3. 讓學習者從設計師角色開始玩遊戲。
4. 玩三次遊戲，每次輪換角色卡。

### 角色：

- 設計師：將數顆寶石和開始箭嘴放在格子上。
- 程式設計員：在同學幫助下，將指令卡放在格子的裡面或旁邊，引導 Byte 接近並收集寶石。
- 測試員：從箭嘴上的 Byte 開始，按照指令在格子內移動 Byte，收集寶石時將寶石加至容器中。如果收集到所有寶石，你可以更新容器內的變數 `numberOfGems`，恭喜你！如果未能收集所有寶石，請全隊一起修正編碼。

### 其他選擇：

如果學習者獨自與你進行活動或在家學習，則可以使用可下載的其他 Keynote 活動自行玩此遊戲。

### 導師教材：

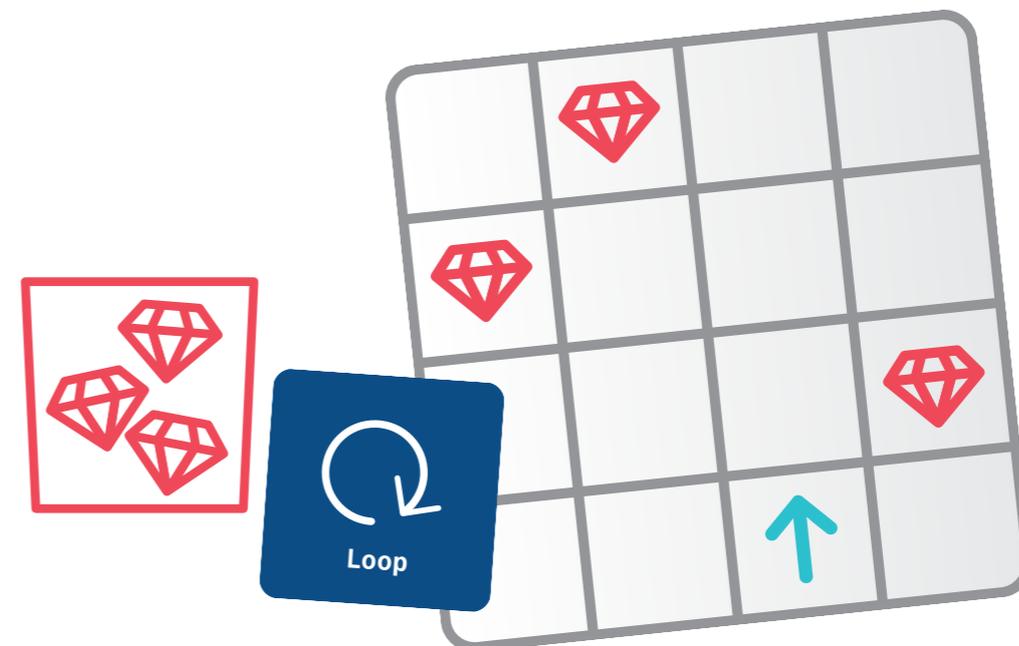
- 畫家膠紙

### 學習者教材：

- 角色卡
- 指令卡：`moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `Loop`
- 寶石
- Byte
- 箭嘴
- 容器，上面標記了：  
`var numberOfGems = _____`
- 畫筆

↓ [下載教材](#)

↓ [下載其他活動選擇](#)



## 探索

**目標：**探索在建立變數時如何使用列表或陣列。

**討論：**如果工作紙要求學習者提供兄弟姐妹的姓名，但學習者有多位兄弟姐妹，會發生什麼事？問問全班的看法。如果學習者建議製作一份列表，告訴他們這就是編碼人員的工作！如果變數有不只一個答案，學習者應該製作一份列表。

讓學習者構思可能有多個答案的問題。

### 例子：

- 朋友的名字 —> Rose、Sam、Joy
- 學習者的年齡 —> 7、8、7、8、7、8、9、7、8、9、8
- 喜愛的顏色 —> 、、、、
- 喜愛的動物 —> 、、、

**要點：**學習者在程式碼建立的列表就像在句子中列出項目一樣。

## 發掘

**目標：**學習者將能夠填入變數，以描述有關自己和拍檔的資料。學習者或許有機會使用陣列作為變數類型。

### 學習者教材：

- 「我的個人資料」和「你的個人資料」工作紙
- 鉛筆
- 顏色筆

### 指示：

1. 學習者完成「我的個人資料」工作紙。
  - 如果學習者有多位兄弟姐妹或多隻寵物，請他們列成清單並以逗號分隔。
2. 將學習者分成兩人一組，以完成「你的個人資料」工作紙。

**其他選擇：**學習者可以使用 iPad 和 Keynote 完成工作紙，拍攝圖片答案的相片並使用格式選項為顏色常值填色。



[下載「個人資料」工作紙](#)

## 遊戲

目標：學習者將能夠識別程式碼中的變數，並探索如何使用包含迴圈的陣列。

### 指示：

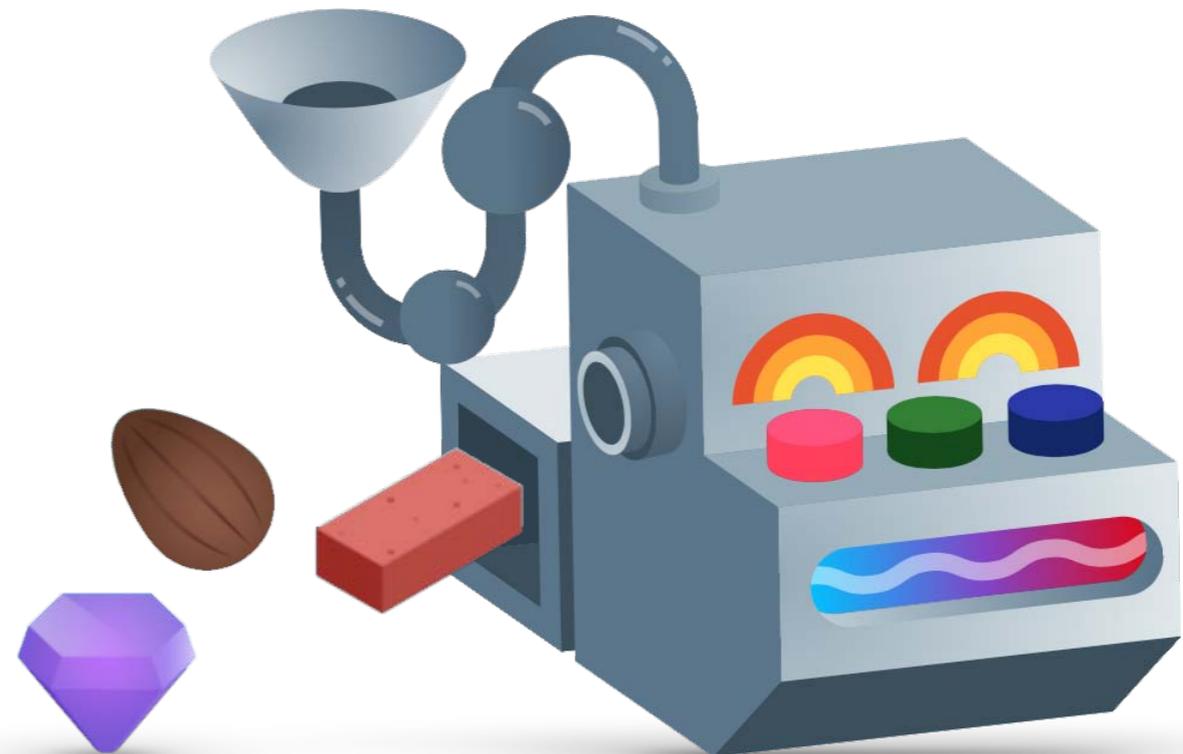
1. 將「程式碼機器」playground 投影到螢幕上。
2. 簡介：
  - 全班一起細閱各頁，必要時停下來解答疑問。
  - 可選：進行首兩頁遊戲，即「探索機器」和「結合顏色」。
3. 使用迴圈：
  - 在這頁，學習者將可把自己的迴圈知識與變數融合起來。
  - 看看學習者能否在使用陣列的程式碼中找到該變數。
  - 按一下或點一下「執行我的程式碼」，了解機器會製作什麼。
  - 繼續前往指示裡的第二步，並更新程式碼以加入第二個變數、項目和嵌套迴圈。再次按一下或點一下「執行我的程式碼」，了解機器會製作什麼。
  - 注意：與學習者進行活動前，先自行體驗此頁面。



程式碼機器

### 導師教材：

- iPad 或 Mac
- Swift Playgrounds app
- 「程式碼機器」playground
- 投影機或顯示器



# App 設計



## 探索

**目標：**在各種裝置上探索熟悉的 app。

**指示：**就學習者在家中或學校使用的 iPad app 展開討論。然後，談論自己、父母或監護人在家中裝置上使用的 app。

**要點：**強調 app 不止存在於手機上，還可在手錶、平板電腦、電腦甚至電視上使用。

**延伸學習：**深入了解幾個 app 例子，詢問學習者這款 app 適合哪些人使用，用途是什麼以及製作動機是什麼。

**例子：**

- App: Swift Playgrounds
- 適合哪些人使用：想要了解 Swift 的人
- 用途是什麼：透過關卡和課程幫助人們學習如何編碼
- 製作動機是什麼：教授沒有或只有少許程式設計知識的人如何編碼

## 發掘

**目標：**分析一款耳熟能詳的 app，讓學習者作好自行設計 app 的準備。

**學習者教材：**

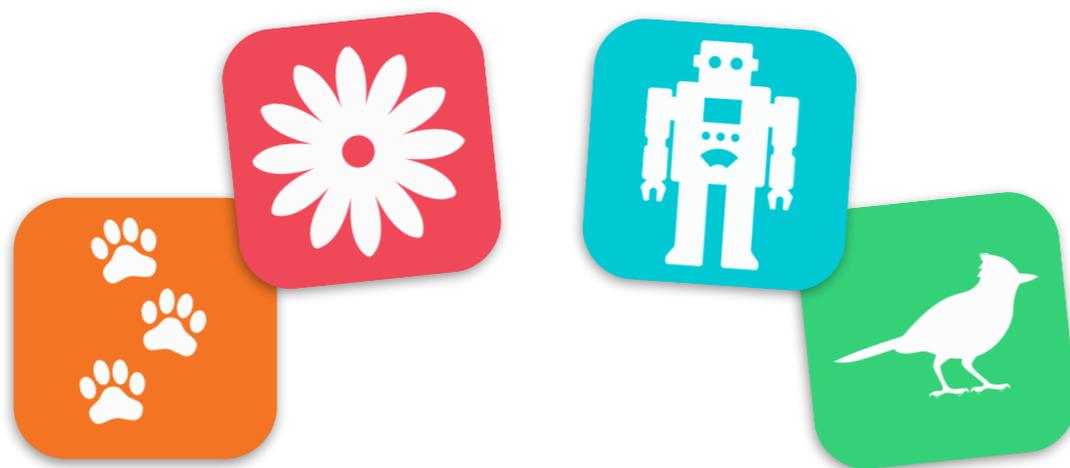
- iPad 裝置
- 「什麼是 App?」工作紙
- 鉛筆
- 彩色畫筆或顏色筆

**指示：**

1. 將學習者分成小組，或者讓他們獨自進行活動。
2. 請學習者選擇一個 iPad app。
3. 讓學習者使用「什麼是 App?」工作紙，以指引他們探索 app。
4. 邀請學習者與全組或拍檔分享自己關於 app 的發現。

**導師貼士：**學習者年紀越小，完成此工作紙時就需要越多幫助。對於幼稚園至一年級學習者的小組，不妨全組一起探索兩至三個 app。

 [下載「什麼是 App?」工作紙](#)



## 遊戲

目標：學習者設計自己的 app！

學習者教材：

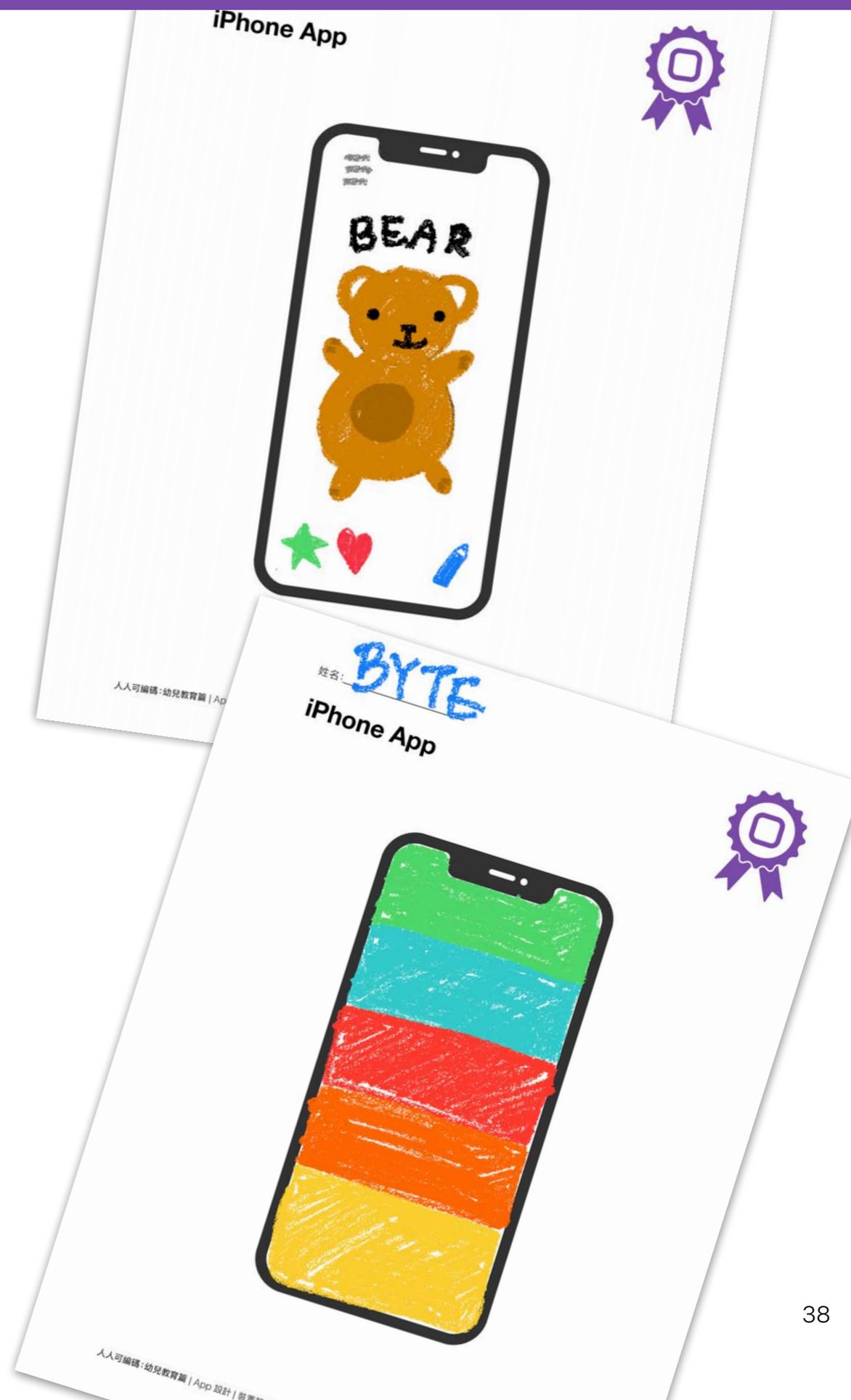
- 「我的 App 設計」工作紙
- 裝置範本
- 額外的紙張
- 鉛筆
- 彩色畫筆或顏色筆

指示：

1. 將學習者分成小組，或者讓他們獨自進行活動。
2. 向學習者完整介紹「我的 App 設計」工作紙，指引他們完成設計 app 的初始程序。
3. 請學習者使用額外的紙張或裝置範本製作 app 原型頁面。
4. 指示學習者使用裝置範本建立 app 原型的最終版本。
5. 邀請每位或每組學習者向所有人展示構思。

↓ 下載「我的 App 設計」工作紙

↓ 下載裝置範本



# 導師資源



## 詞彙表

- **Boolean (布林值)**: 值為 true 或 false 的類型
- **Bug (錯誤)**: 程式碼中的錯誤
- **Command (指令)**: 告訴應用程式執行特定動作的程式碼
- **Data (數據)**: 資料
- **Debug (除錯)**: 找出並修正程式碼中的錯誤
- **Function (函數)**: 可在需要時執行的一組已命名指令
- **Loop (迴圈)**: 重複特定次數的一段程式碼
- **Modify (修改)**: 即改變
- **Sequence (序列)**: 事情發生的次序
- **Step (步驟)**: 較大程序裡的一個動作
- **Toggle (切換)**: 開啟或關閉
- **Variable (變數)**: 具名容器, 用於存放可變值

## CSTA 標準 >

### 1A-AP

- 1A-AP-08: 制定並遵從演算法 (多組逐步指引) 來完成任務, 從而建立日常流程的模型。
- 1A-AP-09: 使用數字或其他符號來表示資料, 從而建立程式儲存和處理數據方式的模型。
- 1A-AP-10: 開發含有序列和簡易迴圈的程式, 以表達想法或解決問題。
- 1A-AP-11: 將解決問題所需步驟拆解 (分拆) 為精確的指示序列。
- 1A-AP-12: 制定計劃以描述程式的事件、目標和預期成果序列。
- 1A-AP-14: 為演算法或程式中的錯誤除錯 (即識別並修正錯誤), 這類演算法或程式包含序列和簡易迴圈。

### 1A-CS

- 1A-CS-01: 選擇並運用合適的軟件以執行各種任務, 同時了解用戶對於使用的科技有不同需求和偏好。

### 1B-AP

- 1B-AP-09: 建立使用變數儲存和修改數據的程式。
- 1B-AP-10: 建立包含序列、事件、迴圈和條件碼的程式。
- 1B-AP-16: 在程式開發的設計、實施和審查階段與同學合作時, 在導師的指導下承擔不同角色。

## 答案範例

接下來的幾頁為每個 Swift Playgrounds 關卡提供其中一種可行的解決方案，但這些關卡均可透過多種方式解決。請鼓勵學習者嘗試不同的方式來指引 Byte 或其他角色。

你可對學習者可能擁有的全部編碼類型和目標表示讚許。有些學習者除了收集寶石，可能還想探索所有關卡，有些則可能想在收集寶石途中盡可能旋轉最多次。別忘了，編碼本應充滿樂趣！



### 學習程式設計 1

#### 「指令」章節

發出指令

```
moveForward()
moveForward()
moveForward()
collectGem()
```

#### 「指令」章節

加入新指令

```
moveForward()
moveForward()
turnLeft()
moveForward()
moveForward()
collectGem()
```

#### 「函數」章節

組合新的動作

```
moveForward()
moveForward()
moveForward()
turnLeft()
turnLeft()
turnLeft()
moveForward()
moveForward()
moveForward()
collectGem()
```

#### 「函數」章節

建立新函數

```
func turnRight() {
    turnLeft()
    turnLeft()
    turnLeft()
}

moveForward()
turnLeft()
moveForward()
turnRight()
moveForward()
turnRight()
moveForward()
turnLeft()
moveForward()
toggleSwitch()
```



學習程式設計 1

「函數」章節

收集、切換、重複

```
func collectToggle() {
  moveForward()
  collectGem()
  moveForward()
  toggleSwitch()
  moveForward()
}
```

```
collectToggle()
turnLeft()
collectToggle()
moveForward()
turnLeft()
collectToggle()
turnLeft()
collectToggle()
```

「迴圈」章節

使用迴圈

```
for i in 1 ... 5 {
  moveForward()
  moveForward()
  collectGem()
  moveForward()
}
```

「迴圈」章節

迴圈每一側

```
for i in 1 ... 4 {
  moveForward()
  collectGem()
  moveForward()
  moveForward()
  moveForward()
  turnRight()
}
```

「迴圈」章節

走到邊緣再返回

```
for i in 1 ... 4 {
  moveForward()
  moveForward()
  toggleSwitch()
  turnLeft()
  turnLeft()
  moveForward()
  moveForward()
  turnLeft()
}
```



學習程式設計 2

「變數」章節

持續追蹤

```
var gemCounter = 0
moveForward()
moveForward()
collectGem()
gemCounter += 1
```



剪刀、石頭、布

遊戲範例

此頁並無解決方案範例，因為遊戲完全可自訂，你可以按自己喜歡的任何方式來玩！



程式碼機器

使用迴圈

```
var colors = [Light.red, Light.green, Light.blue]

var items = [Item.metal, Item.stone, Item.cloth, Item.dirt, Item.DNA, Item.spring, Item.wire, Item.egg, Item.tree, Item.gear, Item.seed, Item.crystal, Item.mushroom, Item.unidentifiedLifeForm]

for item in items {
  setItemA(item)
  setItemB(.dirt)
  switchLightOn(.green)
  forgeItems()
}
```



玩 MeeBot 學編舞

基本舞步

```
bendAndTwist()
happy()
moveBackward()
shake()
skip()
split()
swagger()
twist()
```

舞蹈循環

```
for i in 1 ... 5 {
  bend()
  bend(beats: 2)
  bendAndTwist()
  moveBackward(beats: 9)
}
```

## 更多資源



### 人人可編碼：解謎闖關和探險闖關教師指南

教師指南可讓教育工作者在教授編碼時更有信心，透過真實世界的互動、溝通、團隊合作、明辨性思考及個人化學習來加強學員學習的深度。這些指南還提供了區分課堂活動的評估構思和貼士。[下載《解謎闖關》和《探險闖關》書籍 >](#)

### Apple Teacher 計劃

Apple Teacher 是一項自訂進度的免費專業學習計劃，提供可無限制取用的學習材料和內容，讓你將 Apple 科技應用在教育中。[進一步了解 >](#)

### Apple 專業學習

Apple 專業學習專家透過帶領引人入勝的學習體驗和親身實踐，協助教育工作者發展出創新的指導方法，令學生更投入深度學習。如需進一步了解，請傳送電郵至 [aplshk@apple.com](mailto:aplshk@apple.com)。



### 人人可編碼：Swift 編碼學會

「Swift 編碼學會」是在課餘活動、夏令營或其他非正式學習環境中介紹編碼的上佳方式。「Swift 編碼學會」的單元設計，能完美地迎合編碼新手和較有經驗的人。[下載「Swift 編碼學會」套件 >](#)



### 程式設計快速入門

這份指南提供 10 種有趣好玩的編碼活動，適合 10 歲或以上的兒童參與。學生透過 iPad 及 Mac 版免費 Swift Playgrounds app，在學校或家中都可以學習編碼。[下載《程式設計快速入門》>](#)



### App 設計日誌

學生可以使用《App 設計日誌》來實踐 app 設計流程，以解決學校或社區中的問題。設計日誌鼓勵學生集體構思自己的 app，並為此進行規劃、製作原型和評估，最後還需要宣傳講解自己的 app 原型。[下載《App 設計日誌》>](#)

