

人人可編碼

幼齡學童教育篇

教師指南



目錄

簡介

指令

- 第 1 課：日常作息
- 第 2 課：故事順序
- 第 3 課：舞蹈動作

函數

- 第 1 課：紙寶石
- 第 2 課：小小音樂祭
- 第 3 課：我的緩和情緒函數

迴圈

- 第 1 課：重複繪製花瓣
- 第 2 課：障礙路徑設計
- 第 3 課：擊鼓模式

變數

- 第 1 課：沉下去或浮起來
- 第 2 課：文字遊戲
- 第 3 課：關於我的一切

App 設計

輔導員資源





《人人可編碼：幼齡學童教育篇》專為協助教育工作者和家庭而設計，目的是在低年級學生剛開始發展電腦運算思維技能的時候，向他們介紹程式設計。在這些課程中，幼兒園至 3 年級的學童可以透過探索、發現和邊玩邊學，為程式碼的核心概念奠定基礎。

教學設計

本指南分為四個單元，活動最終的 app 設計計畫案也是如此。每個單元包含三堂課程，每堂課程只著重於介紹一個程式設計的相關概念。每堂課程包含三個活動：「探索」、「發現」及「邊玩邊學」。活動可分成多堂課或多天來進行。

第 1 天：討論與實作學習

探索

介紹與討論程式設計的概念

發現

透過創意活動來熟悉這個概念

約
25 分鐘

邊玩邊學

- 在 Swift Playgrounds app 中和 Byte 一起程式設計
- 透過隨附的學習單和 Keynote 活動來練習編寫程式碼
- 透過不插電的地板闖關程式設計遊戲，在現實世界中重現 Byte 的世界

約
25 分鐘

第 2 天：將學習與程式設計做連結

範圍及順序

本指南專為幼兒園至 3 年級的學童所設計，包含四個單元並可依任意順序完成。我們鼓勵你隨時運用「App 設計」單元，甚至可以在一學年中多次使用，以配合學習者日益增長的程式碼和 app 知識。

範例：

年級	單元	最終計畫案	約略時間總計
幼兒園	指令	App 設計	4 小時
1 年級	函數	App 設計	4 小時
2 年級	迴圈	App 設計	4 小時
3 年級	變數	App 設計	4 小時

持續學習

對於適用於 4 到 8 年級學生的教學，《人人可編碼：解謎闖關》和《人人可編碼：探險闖關》課程指南結合《App 設計日誌》和《App 發表活動指南》，可提供超過 90 個小時的學習。進一步了解 [《人人可編碼課程指南》](#)。



學習者作品集 (選用)

收集每個單元中完成活動的作品，與學習者一起製作作品集。



單元	課程	建議成品
指令	日常作息	<ul style="list-style-type: none"> 《發出指令》學習單 《加入新指令》學習單
	故事順序	<ul style="list-style-type: none"> 故事順序的故事橋段圖片 故事順序的組圖
	舞蹈動作	<ul style="list-style-type: none"> 舞蹈動作卡 舞蹈動作影片 (選用)
函數	紙寶石	<ul style="list-style-type: none"> 紙寶石形狀 《組合新的動作》學習單 《建立新函數》學習單
	小小音樂祭	<ul style="list-style-type: none"> 小小音樂祭的音樂會影片或編寫的函數
	我的緩和情緒函數	<ul style="list-style-type: none"> 我的緩和情緒函數之繪圖或影片 《收集、切換、重複》學習單
迴圈	重複繪製花瓣	<ul style="list-style-type: none"> 重複繪製花瓣 《使用迴圈》學習單 《迴圈每一側》學習單
	障礙路徑	<ul style="list-style-type: none"> 障礙路徑的影片或照片 (選用)
	循環擊鼓	<ul style="list-style-type: none"> 《走到邊緣再返回》學習單 循環擊鼓的影片或照片 (選用)
變數	沉下去或浮起來	<ul style="list-style-type: none"> 沉下去或浮起來 《持續追蹤》學習單
	文字遊戲	<ul style="list-style-type: none"> 文字遊戲
	關於我的一切	<ul style="list-style-type: none"> 關於我的一切 關於你的一切
App 設計		<ul style="list-style-type: none"> App 是什麼? 我的 App 設計 App 設計原型

在 iPad 或 Mac 上開始使用 Swift Playgrounds



進入課程之前，請確認你已下載 [Swift Playgrounds](#)、[Pages](#)，以及 [Keynote](#)。

本指南中的單元使用了不同的 Playground 組合。以下是各單元所需項目：

單元	Playground	如何在 Swift Playgrounds 中下載
指令	 學習程式設計 1  玩 MeeBot 學編舞	<p>如果要訂閱 MeeBot Playgrounds Feed，請在「更多 Playgrounds」畫面捲動到最下方，然後點一下「輸入訂閱 URL」。接著輸入：ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json。</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>如果要訂閱 MeeBot Playgrounds Feed，請在「更多 Playgrounds」畫面捲動到最下方，然後點一下「輸入訂閱 URL」。接著輸入：ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json。</p>
變數	 學習程式設計 2  剪刀、石頭、布  程式碼機器	<p>「剪刀、石頭、布」和「程式碼機器」可在「更多 Playgrounds」畫面的「書籍」區塊中找到。</p>
App 設計		

請前往 [App Store](#) 查看 Swift Playgrounds 的最低需求。請前往 [Apple 支援](#) 以取得 Swift Playgrounds 的協助。

輔導員提示

請試用這些提示，與學習者一起充分發揮課程效益。

探索和發現活動：

- 在編寫和顯示程式碼的時候，簡化所有語法和大小寫，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 = ■`

邊玩邊學活動：

- 為了讓幼齡學童更容易操作 Swift Playgrounds app，請遵循課程計畫中的指示。其中包括：
 - 大家一起閱讀簡介
 - 簡化隨附學習單的指引，讓學習者有機會提出自己的解決方案
 - 使用一部輔導員專屬 iPad 或 Mac 來突破 app 中的關卡
- `let` 和 `var`：本指南沒有包含關鍵字 `let` 的說明。為了避免在 Swift Playgrounds app 中產生混淆，將頁面展示給學習者前，請將所有關鍵字 `let` 改成 `var`。在我們建議的 Playground 中，這兩個關鍵字可以互相替換。
 - `let` = 變數不會改變
 - `var` = 變數會改變

延伸活動：

- 提供更多在地板上進行的「邊學邊玩」活動，加入數學、讀寫、常用字、拼寫和其他內容。嘗試「函數」單元中「邊玩邊學」的地板活動來獲得靈感。
- 請學習者製作專屬的指令卡 (例如 `twirl()` 或 `jump()`)，以自訂在地板上進行的「邊學邊玩」活動。



簡介頁面



Playground 頁面



指令



概覽

第 1 課：日常作息

- 探索：討論如何將烘焙與指令建立關聯
- 發現：日常作息活動
- 邊玩邊學：發出指令和加入新指令

第 2 課：故事順序

- 探索：討論故事情節順序如何與指令建立關聯
- 發現：故事順序活動
- 邊玩邊學：地板闖關遊戲

第 3 課：舞蹈動作

- 探索：討論舞蹈動作如何與指令建立關聯
- 發現：舞蹈動作活動
- 邊玩邊學：你好 MeeBot 和基礎舞步

學習者的收穫

- 利用日常生活中的範例來說明逐步指示
- 以合理方式依序排列指示
- 為指示和程式碼進行測試與除錯

詞彙

- **Sequence (序列)**：事情發生的順序
- **Step (步驟)**：一連串程序中的一個動作
- **Modify (修改)**：做出改變
- **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)**：找到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：使用烘焙布朗尼蛋糕的比喻來介紹指令概念。

討論：

- 他們是否要參照巧克力布朗尼的食譜？
- 他們是否要遵循食譜的步驟順序？

學習重點：食譜裡的每個步驟或指示就像是程式碼的指令。請學習者提出想法，編寫自己專屬的指令。

發現

目標：依序分辨逐步指示，建立日常作息程序的模型。

材料：洗手卡

指引：

1. 將洗手卡洗牌，接著將卡片排列在桌面或白板上。請不要依照順序排列卡片。
2. 詢問學習者是否覺得洗手順序有誤。
3. 請學習者一次移動一張卡片到正確位置，針對洗手指示進行除錯，也就是修正。

替代活動：

將學習者分為兩人以上的小組，每組發一套卡片。

延伸活動：

請學習者提出一件每天都會做的事，接著列出這件事的逐步指示，並為特定步驟製作圖片。

↓ [下載《洗手卡》](#)



邊玩邊學

目標：在 Swift Playgrounds app 的「學習程式設計 1」中，學習者能在加入指令時以正確的順序排列，收集他們的第一顆寶石。

指引：

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：
 - 全班一起閱讀章節內容，若有疑問隨時暫停並討論。
3. 發出指令：
 - 為了讓 Byte 移動到有寶石的地方，請幫學習者複習須要用到的兩個指令：`moveForward()` 和 `collectGem()`。
 - 請學習者嘗試以不同的方式，讓 Byte 從開始箭頭處移動到有寶石的地方並收集寶石。請大家將指令記錄在學習單或另一張紙上。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼來完成關卡。點一下「執行我的程式碼」。
 - 嘗試多個不同的構想。
 - 與 Byte 一起慶祝！

延伸活動：

當學習者準備好時，便可前往下一頁進行「加入新指令」。此時，學習者將使用新的指令 `turnLeft()`。



學習程式設計 1

輔導員所需材料：

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

學習者所需材料：

- 《發出指令》和《加入新指令》學習單
- 鉛筆
- 額外的紙張 (選用)



[下載學習程式設計的學習單](#)



探索

目標：探索書中故事如何依照順序進行（從開頭到中段再到結尾），使故事情節有條理。

討論：

- 請學習者思考，書籍內容是否有一個特定的順序。
- 如果書籍的開頭、中段、結尾不照順序排列，會發生什麼事情？
- 探索數個範例。

學習重點：幫助學習者理解程式設計，強調依正確順序向程式碼下達指令的重要性，就像故事中的橋段一樣。



發現

目標：在為多種不同的故事橋段製作圖片後，學習者將能依序排列圖片，正確地重現故事。

輔導員所需材料：

- 白板
- 麥克筆

學習者所需材料：

- 紙
- 麥克筆或彩色鉛筆
- 替代活動：iPad 裝置和一款繪圖 app

指引：

1. 為學習者閱讀一篇耳熟能詳的故事。全班一起決定主要的故事橋段。在理想情況下，請提出四至六個故事橋段。
2. 將學習者分成小組，每組人數應與故事橋段個數相對應，例如：四個故事橋段就要四人一組。
3. 請每位組員畫出一個故事橋段。
4. 每組輪流上台，請組員不按順序站在台前，同時展示手中的圖片。
5. 請台下觀眾一次移動一張圖片來重新排序。
6. 將學習者排列出正確順序後，為每個小組拍攝照片。

延伸或替代活動：

請每組選擇不同故事，然後全組一起決定故事橋段，再開始畫圖片。

邊玩邊學

目標：學習者將能利用指向性指令，引導 Byte 在實體網格上移動到有寶石的地方。

準備：將學習者分成三人一組。使用油漆用遮蔽膠帶，在地板上為每組建立 4 × 4 的網格。

指引：

1. 分發材料，接著將學習者分成三人一組。
2. 閱讀各角色卡，接著將角色指派給每個組員以進行第一場遊戲。
3. 請學習者從設計師角色開始來玩這個遊戲。
4. 玩三次，每次輪流使用不同的角色卡。

角色：

- 設計師：在網格上放置寶石和開始箭頭。
- 程式設計師：在夥伴的協助下，將指令卡放置於網格上方或旁邊，以引導 Byte 移動到有寶石的地方並收集寶石。
- 測試人員：讓 Byte 從箭頭處開始，照著指令卡引導 Byte 在網格上移動。如果成功收集到寶石，恭喜你！如果沒有，請與小組一起除錯或修正程式碼。

替代活動：

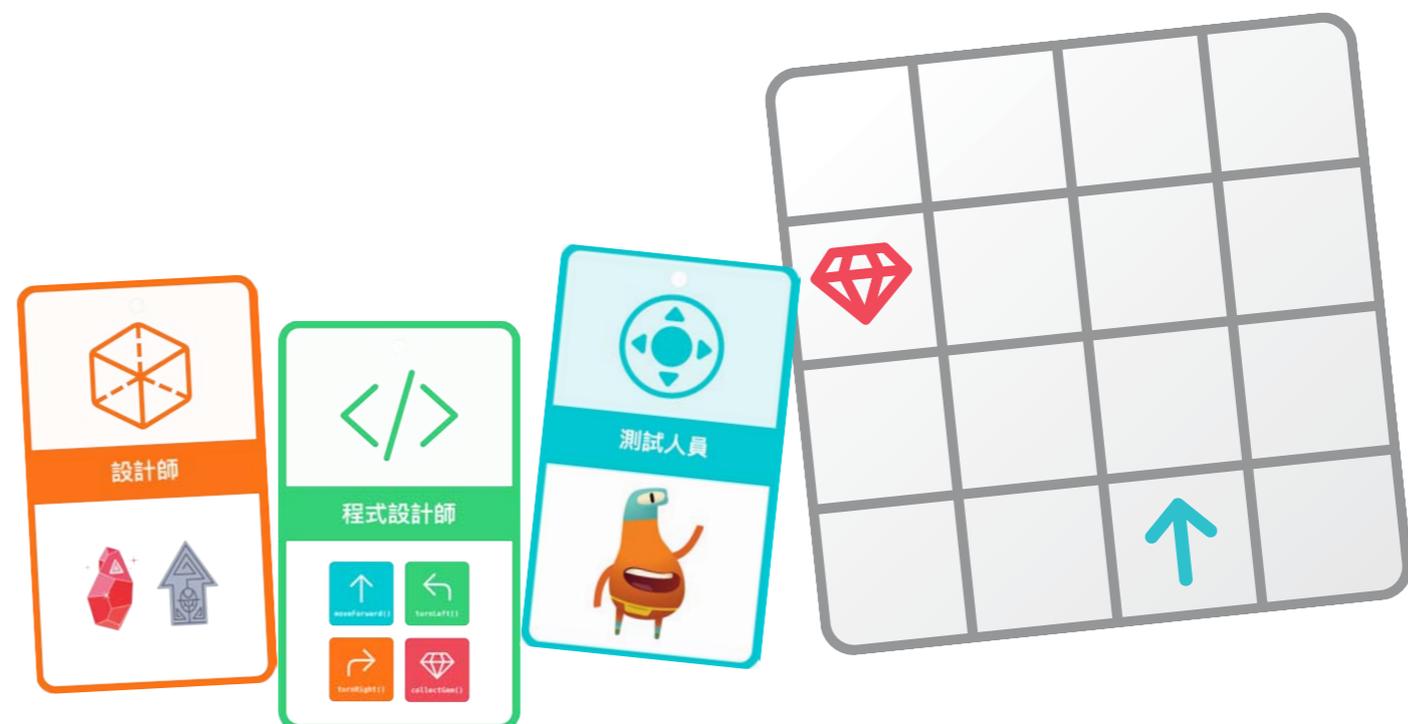
如果學習者是與你進行一對一課程或在家自學，則可以自行下載 Keynote 替代活動來進行這個遊戲。

輔導員所需材料：

- 油漆用遮蔽膠帶

學習者所需材料：

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



探索

目標：讓學習者了解，程式設計可以充滿創意！

討論：

- 詢問學習者是否曾經學過任何舞蹈。
- 這段舞蹈是否要按照一定的舞步順序？
- 他們如何知道下一個舞步？
- 這些舞蹈動作有名稱嗎？
- 在同一支舞蹈或不同的舞步組合裡，學習者是否曾經使用相同的舞蹈動作？

學習重點：幫助學習者理解程式設計就像編排舞蹈一樣充滿創意，程式設計者可以建立新的指令，並以不同又有趣的方式組合在一起。

發現

目標：編排一段簡短的舞步組合，並搭配卡片來呈現舞蹈動作。每張「舞蹈動作」卡就像是「學習程式設計」Playground 中的指令。

學習者所需材料：

- iPad 裝置
- Keynote app
- 「相機」app
- 跳舞的空間

指引：

1. 請學習者分為兩人以上的小組，編排一段簡短的舞步組合。
2. 學習者編排好舞步組合後，請他們製作各式舞蹈動作卡。每張卡片上應包含塗鴉和舞步名稱，請學習者盡情發揮創意，就算有些無厘頭也沒關係。
3. 每組輪流表演自己的舞蹈，然後全班一起來場熱舞派對！

替代活動：

學習者可以下載下方的舞蹈動作卡來編排自己的舞步，或將其作為參考以製作自己的卡片。

延伸活動：

學習者製作自己的舞蹈影片並展示給全班看。



[下載舞蹈動作卡](#)



邊玩邊學

目標：建立步驟序列來教導 MeeBot 機器人新的舞步。

指引：

1. 將「玩 MeeBot 學編舞」的 Playground 投影到螢幕上。如果你沒有訂閱此 Playground，須訂閱才能取得。
2. 簡介：
 - 全班一起閱讀章節內容，若有疑問隨時暫停並討論。
3. 你好 MeeBot：
 - 點一下「執行我的程式碼」並觀看機器人跳舞。
4. 基礎舞步：
 - 將學習者分為兩人以上的小組，或讓他們個別在自己的 iPad 上，從建議列表選擇八個指令並觀看機器人跳舞。
 - 請學習者分享他們的舞蹈動作，或讓全班一起設計幾個不同的舞蹈動作。
 - 與機器人共舞！

延伸活動：

- 前往下一頁的「舞步組合」，學習者可以在 `myDanceRoutine()` 函數內加入舞步，指令數量不限。



玩 MeeBot 學編舞

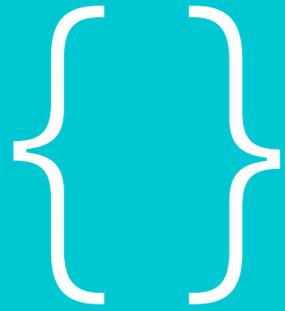
輔導員所需材料：

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

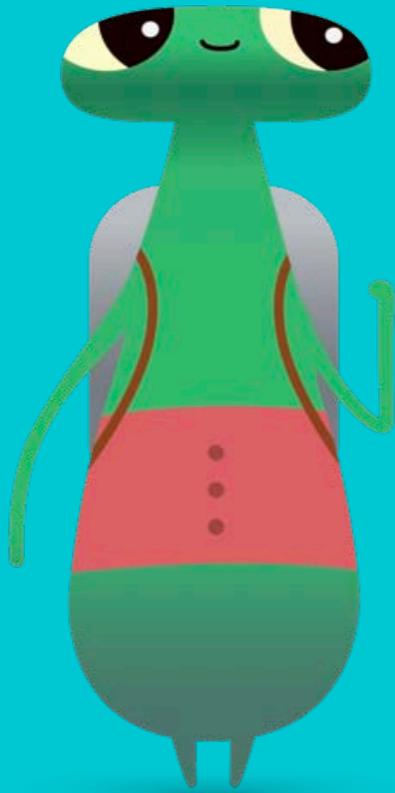
學習者所需材料：

- iPad 裝置 (選用)





函數



概覽

第 1 課：紙寶石

- 探索：進行逐步指示的討論
- 發現：紙寶石活動
- 邊玩邊學：組合新的動作和建立新函數

第 2 課：小小音樂祭

- 探索：討論命名函數的方式
- 發現：小小音樂祭活動
- 邊玩邊學：地板闖關遊戲

第 3 課：我的緩和情緒函數

- 探索：討論解決問題的多種方法
- 發現：我的緩和情緒函數活動
- 邊玩邊學：收集、切換、重複

學習者的收穫

- 將大問題或任務分解成較小的步驟
- 建立一系列步驟來解決問題或完成任務
- 為函數命名
- 為程式碼進行測試與除錯

詞彙

- **Function (函數)**：一組具名的指令，能在需要時執行
- **Toggle (切換)**：打開或關閉

標準

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

探索

目標：探索如何封裝一系列指令並加以命名。

討論：全班一起決定一項日常作息作為活動主軸。請學習者舉出他們的日常作息，以及完成作息的步驟。

範例：睡前習慣

- 步驟 1：刷牙
- 步驟 2：上洗手間
- 步驟 3：閱讀
- 步驟 4：說晚安
- 步驟 5：關燈

學習重點：建立一組指示並加以命名，這與編寫函數是相同的概念。

延伸活動：請學習者思考，是否有任何步驟的指示可以更加明確。例如，刷牙有哪些具體步驟？

發現

目標：一開始，學習者要先遵循指引製作紙寶石，接著他們可選擇自己要製作的形狀，並寫下或畫出製作的指示。

學習者所需材料：

- 紙
- 剪刀
- 鉛筆
- iPad 裝置 (選用)

指引：

向學習者示範製作紙寶石的方式：

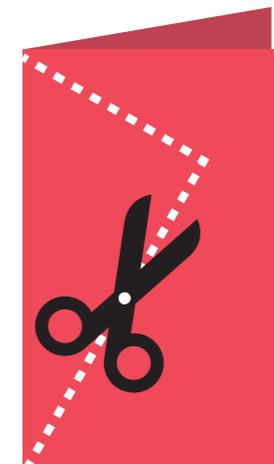
1. 將一張紙對折。
2. 從折線上方的邊角畫一條線，連到中央上方一至二吋的地方。
3. 從上一條線結束的地方，畫另一條線到折線下方的邊角。
4. 沿著剛才畫好的線條剪下。
5. 拿出剪下的紙並打開對折。

請學習者各自製作自己設計的形狀：

1. 將學習者分成小組。
2. 請各小組自己決定要製作的形狀。
3. 預留時間讓學習者練習製作該形狀一至兩次。
4. 請學習者寫下或描繪製作該形狀的指引，然後為該指引命名，如「畫個圓圈」或是「字母 T」。

替代活動：

製作影片為他們示範如何製作形狀。



邊玩邊學

目標：以小組為單位，學習者將學會分解步驟，讓 Byte 移動到有寶石的地方。

指引：

1. 將「學習程式設計 1」的 Playground 投影到螢幕上。前往「學習程式設計 1」的「函數」章節。
2. 簡介：
 - 全班一起閱讀章節內容，若有疑問隨時暫停並討論。
3. 組合新的動作：
 - 複習指令 `moveForward()`、`turnLeft()` 和 `collectGem()`，請記住，你沒有 `turnRight()` 的指令。
 - 請學習者嘗試以不同的方式，讓 Byte 從開始箭頭處移動到有寶石的地方並收集寶石。請大家將指令記錄在學習單或另一張紙上。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼來完成關卡。點一下「執行我的程式碼」。
 - 嘗試多個不同的構想。
 - 與 Byte 一起慶祝！
4. 建立新函數：
 - 請學習者根據在上一個「組合新的動作」Playground 頁面中所學的知識，提出建立 `turnRight()` 函數的想法。
 - 請學習者使用 `turnRight()` 函數，嘗試以不同的方式將 Byte 從開始箭頭處引導到關閉的開關，並將開關打開。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼來完成關卡。點一下「執行我的程式碼」。
 - 嘗試多個不同的構想。
 - 與 Byte 一起慶祝，剛才的關卡並不簡單！



學習程式設計 1

輔導員所需材料：

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

學習者所需材料：

- 《組合新的動作》和《建立新函數》學習單
- 鉛筆
- 額外的紙張 (選用)



[下載學習程式設計的學習單](#)

探索

目標：透過為歌曲提供描述性名稱，應用指令與函數的相關知識。

討論：請學習者提出多首不同的歌曲，並為每首歌曲提供描述性名稱。

範例：以《一閃一閃亮晶晶》這首歌為例，函數呼叫可以是 `singTwinkle()`，但 `singSong1()` 就不是適合的名稱，因為第一首歌可能會換成別首。

學習重點：為函數提供描述性名稱很重要，因為你和其他人可以更容易理解程式碼的內容。

發現

目標：學習者將在音樂會函數中，透過呼叫不同歌曲指令來舉辦一場音樂會。

輔導員所需材料：

- iPad 或 Mac
- 投影機或顯示器
- 白板
- 麥克筆

指引：

1. 幫助學習者為多首歌曲的函數命名，例如 `singHappyBirthday()`。
2. 全組一起決定演唱歌曲的順序。
3. 為音樂會編寫函數定義，並在其中放入歌曲指令。

範例：

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

替代活動：

將學習者分成小組，每組都有自己的專屬歌單、歌曲函數名稱和歌曲演唱順序。接下來，各組表演自選歌曲，並為自己的音樂會製作影片。

邊玩邊學

目標：學習者會先解析一個簡單的方程式，接著將寶石放置在答案上，然後利用方向性指令引導 Byte 在網格上移動。

準備：將學習者分成三人一組。使用油漆用遮蔽膠帶，在地板上為每組建立 4 × 4 的網格。在一個方格內放置開始箭頭，然後在其餘的方格內放置一個數字。

指引：

1. 分發材料，接著將學習者分成三人一組。
2. 閱讀各角色卡，接著將角色指派給每個組員以進行第一場遊戲。
3. 請學習者從設計師角色開始來玩這個遊戲。
4. 玩三次，每次輪流使用不同的角色卡。

角色：

- 設計師：擲兩個骰子。在夥伴的協助下，將所得數字相加，並將寶石放在網格上顯示其總和的方格內。
- 程式設計師：在夥伴的協助下，將指令卡放置於網格上方或旁邊，以引導 Byte 移動到有寶石的地方並收集寶石。
- 測試人員：讓 Byte 從箭頭處開始，照著指令卡引導 Byte 在網格上移動。如果成功收集到寶石，恭喜你！如果沒有，請與小組一起修正程式碼。

替代活動：

如果學習者是與你進行一對一課程或在家自學，則可以自行下載 Keynote 替代活動來進行這個遊戲。

輔導員所需材料：

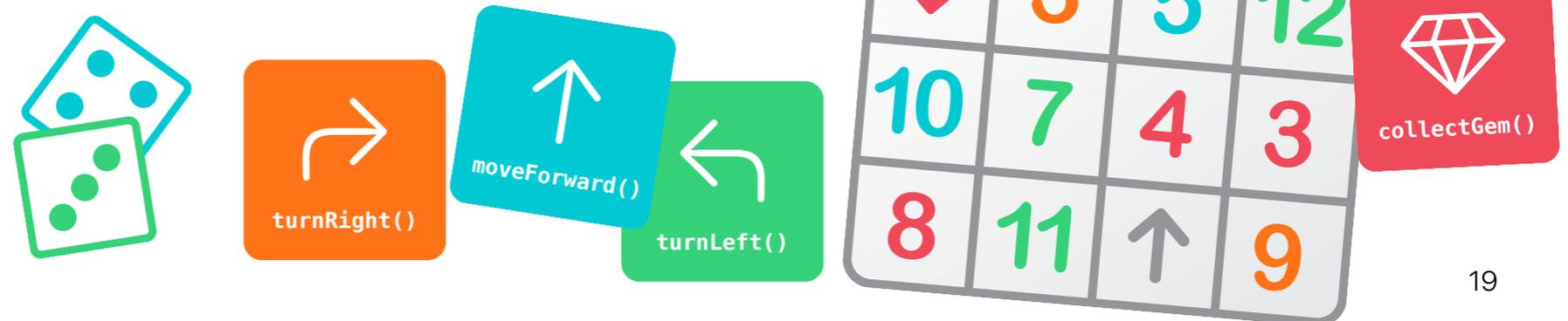
- 油漆用遮蔽膠帶
- 每個網格須有一組印刷數字卡

學習者所需材料：

- 角色卡
- 指令卡：`moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()`
- 寶石
- Byte
- 箭頭
- 兩個骰子

↓ [下載材料](#)

↓ [下載替代活動](#)



探索

目標：學習者將了解通常解決問題的方法不止一種。

討論：請學習者回想曾經遇到的難題，並分享解決方法。詢問是否有人會用不同方法解決。探索多個不同的問題和解決方案。

學習重點：幫助學習者理解程式設計，並了解通常解決程式設計問題的方法不止一種。

發現

目標：學習者編寫一個緩和情緒技巧的函數，並加以命名。

學習者所需材料：

- 「我的緩和情緒函數」學習單
- 鉛筆
- 彩色筆或鉛筆

指引：

提示：如果可以，最好讓學習者個別進行此活動。

1. 請學習者集思廣益，如果在家中或是學校心情不好，有些什麼方法可以讓自己平靜下來。請他們將自己的緩和情緒技巧化為數個步驟。
2. 分發「我的緩和情緒函數」學習單，請學習者畫出他們靜心技巧的步驟。
3. 請學習者為他們的緩和情緒技巧命名。他們可以使用駝峰式大小寫命名法，例如 `countToTen()`，或直接使用短句，如 `Count to ten` (數到十)。

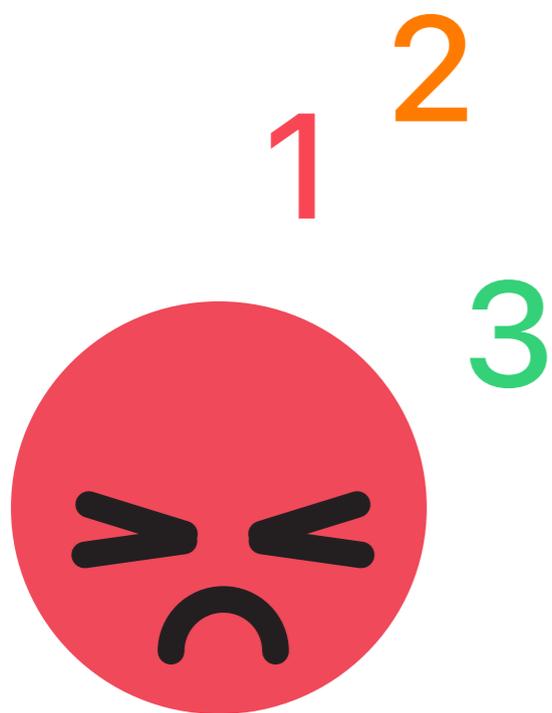
延伸活動：

不插電：請學習者在小組內或全班面前，展現他們的緩和情緒技巧。

使用 iPad：學習者製作影片，與班級分享他們的緩和情緒技巧。



[下載《我的緩和情緒函數》學習單](#)



邊玩邊學

目標：學習者能夠編寫由多個不同類型指令組成的函數，然後使用該函數來完成關卡。

指引：

1. 將「學習程式設計 1」Playground 的「收集、切換、重複」頁面投影到螢幕上，指出需要學習者幫忙完成的空函數。
2. 收集、切換、重複：
 - 複習指令 `moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `toggleSwitch()`。
 - 請學習者識別關卡中重複的部分，然後按照自己的構想完成 app 中的函數，並為之命名。
 - 請學習者為函數創造一個符號，並將符號和函數名稱記錄在學習單的指令表中。
 - 請學習者嘗試以不同的方式並搭配所加入的指令，引導 Byte 收集全部寶石並切換全部開關。
 - 請大家將指令記錄在學習單或另一張紙上。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼來完成關卡。點一下「執行我的程式碼」。
 - 嘗試多個不同的解法。
 - 全班一同分享成就喜悅，剛才的關卡並不簡單！



學習程式設計 1

輔導員所需材料：

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

學習者所需材料：

- 《收集、切換、重複》學習單
- 鉛筆
- 額外的紙張 (選用)



[下載學習程式設計的學習單](#)





迴圈



概覽

第 1 課：重複繪製花瓣

- 探索：討論程式碼中的重複步驟如何與現實生活建立關聯
- 發現：重複繪製花瓣活動
- 邊玩邊學：使用迴圈和迴圈每一側

第 2 課：障礙路徑

- 探索：討論迴圈中的停止點
- 發現：障礙路徑活動
- 邊玩邊學：地板闖關遊戲

第 3 課：擊鼓模式

- 探索：討論音樂中的迴圈
- 發現：擊鼓模式活動
- 邊玩邊學：走到邊緣再返回和舞步循環

學習者的收穫

- 識別程式碼中的迴圈
- 將大問題或任務分解成較小的步驟
- 建立指令序列，並使用迴圈來重複該序列
- 為指示和程式碼進行測試與除錯

詞彙

- **Loop (迴圈)**：會重複執行一定次數的程式碼區塊

標準

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

探索

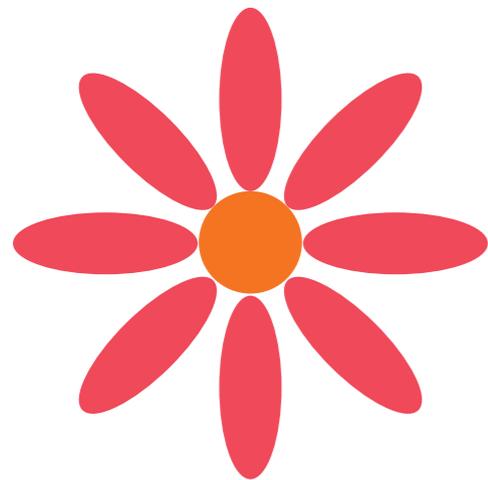
目標：將迴圈概念與現實生活做連結。

討論：探索在現實生活中，學習者可能會重複的工作或步驟。

範例：

- 步行
- 騎自行車
- 縫紉、棒針編織或鉤針編織

學習重點：迴圈會依照你指定的次數重複執行一個或一組指令。



發現

目標：學習者將透過繪製獨一無二的花朵來開始探索迴圈的概念。

學習者所需材料：

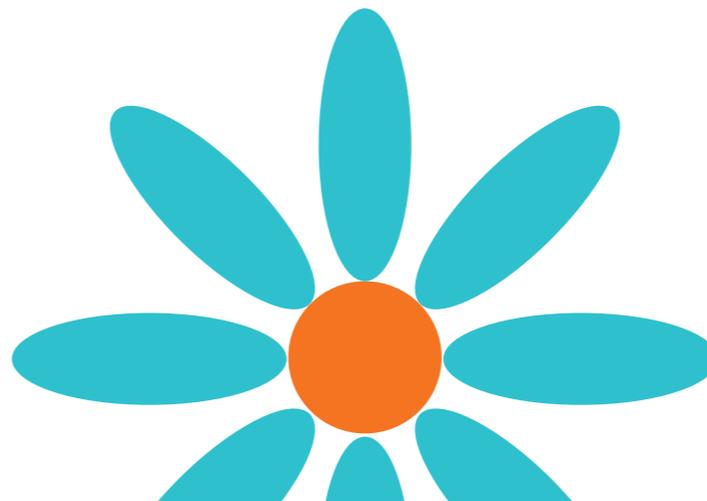
- 「重複繪製花瓣」學習單
- 色紙
- 鉛筆
- 剪刀
- 口紅膠
- 骰子

指引：

1. 學習者在色紙上繪製一片花瓣 (約為手掌長度) 並將其剪下。這將作為用來製作花朵的花瓣樣板。
2. 接著，請每位學習者擲兩個骰子，將所得數字相加，接著在《重複繪製花瓣》學習單，將該數字填入迴圈中的空白處。此數字代表著花朵的花瓣數量。
3. 學習者要使用各自的花瓣樣板，在色紙上描繪出形狀，並為他們的花朵剪出正確花瓣數量。
4. 透過「重複繪製花瓣」學習單，學習者將組合自己的花朵，並將其黏在正確的位置上。



[下載《重複繪製花瓣》學習單](#)



邊玩邊學

目標：學習者將能在迴圈中編寫程式碼來收集全部的寶石。

指引：

1. 將「學習程式設計 1」Playground 中的「For 迴圈」章節簡介頁面投影到螢幕上。
2. 簡介：
 - 全班一起閱讀章節內容，若有疑問隨時暫停並討論。
3. 使用迴圈：
 - 為學習者示範傳送門的運作方式，並複習指令 `moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()`。
 - 請學習者嘗試以不同的方式，引導 Byte 從開始箭頭處移動到有寶石的地方並收集寶石，看看有哪些重複的指令。請大家將指令記錄在學習單或另一張紙上。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼，引導 Byte 收集第一顆寶石並走到傳送門。
 - 詢問學習者共有多少顆寶石，接著將回答的數字加入迴圈中。點一下「執行我的程式碼」。
 - 嘗試多個不同的解法。
 - 與 Byte 一起慶祝！
4. 迴圈每一側：
 - 請學習者嘗試以不同的方式，引導 Byte 收集到全部的寶石，並查看有哪些重複的指令。
 - 若要加入 `for` 迴圈，可以選擇使用編輯器底端的程式碼建議，或點一下螢幕上方的加號 `+`。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼來完成關卡。點一下「執行我的程式碼」。
 - 嘗試多個不同的構想。
 - 與 Byte 一起慶祝！



學習程式設計 1

輔導員所需材料：

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

學習者所需材料：

- 《使用迴圈》和《迴圈每一側》學習單
- 鉛筆
- 額外的紙張 (選用)



[下載學習程式設計的學習單](#)

探索

目標：探索迴圈一定要有特定結束點的原因。

討論：請學習者想像一下摩天輪，或是他們熟悉的其他遊樂設施。如果在旋轉了五圈後，操作員沒有按下停止按鈕，會發生什麼事情？請學習者提出其他範例，如果迴圈不停止，會發生什麼事情？

學習重點：幫助學習者了解，如果他們不結束迴圈，迴圈便會無限重複。

發現

目標：學習者將藉由通過自己設計的障礙路徑，了解迴圈的運作方式。

材料：

- 體能活動的空間
- 障礙路徑道具
- 骰子

指引：

1. 在教室或室外設計簡短的障礙路徑。
2. 擲一顆骰子，並請學習者根據所得數字來決定重複障礙路徑的次數。

替代活動：

請學習者想出一系列的動作，例如碰觸你的腳趾、跳躍和單腳踢腿。擲一顆骰子，請學習者根據所得數字來決定重複這一系列動作的次數。



邊玩邊學

目標：學習者將能建立有重複模式的關卡，然後全組一起闖關。

準備：將學習者分成三人一組。使用油漆用遮蔽膠帶，在地板上為每組建立 4 × 4 的網格。

指引：

1. 分發材料，接著將學習者分成三人一組。
2. 閱讀各角色卡，接著將角色指派給每個組員以進行第一場遊戲。
3. 請學習者從設計師角色開始來玩這個遊戲。
4. 玩三次，每次輪流使用不同的角色卡。

角色：

- 設計師：在夥伴的協助下，將三顆寶石以重複形式放在網格上。在網格上放置開始箭頭。
- 程式設計師：在夥伴的協助下，將指令卡放置於網格的上方或旁邊，以引導 Byte 移動到有寶石的地方並收集寶石。使用 Loop (迴圈) 卡，告訴測試人員重複指令的次數。
- 測試人員：讓 Byte 從箭頭處開始，照著指令卡引導 Byte 在網格上移動。如果成功收集到全部寶石，恭喜你！如果沒有，請與小組一起修正程式碼。

替代活動：

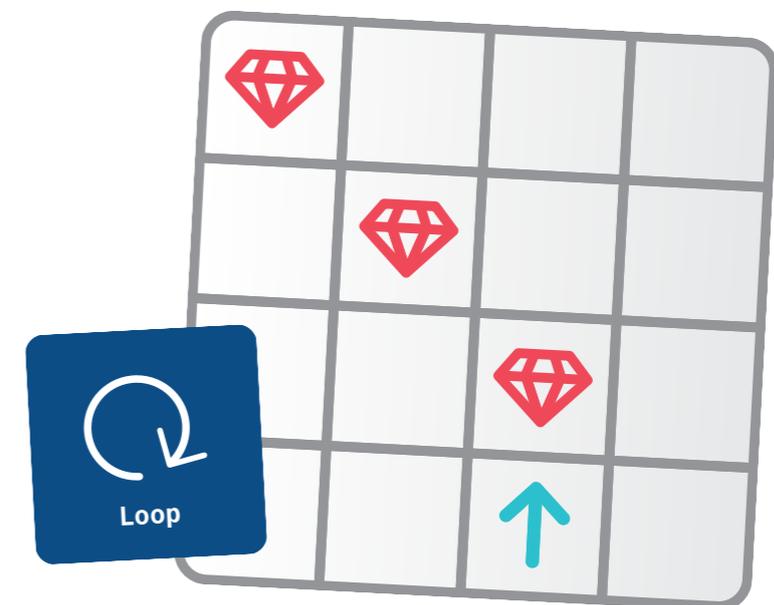
如果學習者是與你進行一對一課程或在家自學，則可以自行下載 Keynote 替代活動來進行這個遊戲。

輔導員所需材料：

- 油漆用遮蔽膠帶

學習者所需材料：

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



探索

目標：探索音樂中的重複模式。

討論：請學習者分享曾經演奏過的樂器或唱過的歌曲。詢問他們在演奏或演唱時，是否曾重複節奏或是副歌。他們能否舉出歌曲或音樂中還有哪些地方重複？

學習重點：加強概念，表示迴圈由兩個部分組成：

- 指令
- 重複執行的次數

發現

目標：學習者將能重複一段鼓聲模式，將重覆執行程式碼的概念與現實生活做連結。

材料：

- 可打擊的東西，如地板、大腿或書本
- 可圍成一圈坐下的空間

指引：

1. 請學習者圍成一圈坐下。
2. 請學習者重複你創作的鼓聲節奏，並比出手勢決定次數。例如：當你比出四根手指的時候，學習者應重複四次鼓聲節奏，然後停止。
3. 為確保每個人都有機會帶領大家，可請圓圈中的學習者輪流擔任首席鼓手，也可以小組形式進行。

延伸活動：

請學習者製作鼓組。



邊玩邊學

目標：學習者將在迴圈中呼叫出多個不同指令，並決定呼叫迴圈的次數。

指引：

1. 將「學習程式設計 1」的 Playground 投影到螢幕上。前往「For 迴圈」章節的第三頁「走到邊緣再返回」。
2. 走到邊緣再返回：
 - 複習指令 `moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `toggleSwitch()`。
 - 請學習者嘗試以不同的方式，讓 Byte 從開始箭頭處引導到關閉的開關，並將開關打開。
 - 若要加入 `for` 迴圈，可以選擇使用編輯器底端的程式碼建議，或點一下螢幕上方的加號 +。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼來完成關卡。點一下「執行我的程式碼」。
 - 嘗試多個不同的構想。
 - 與 Byte 一起慶祝！
3. 離開「學習程式設計 1」並打開「玩 MeeBot 學編舞」的 Playground，前往「舞步循環」頁面。（此 Playground 頁面沒有相對應的學習單。）
4. 舞步循環：
 - 將學習者分為兩人以上的小組，或讓他們個別在自己的 iPad 上完成迴圈，並觀看機器人跳舞。
 - 請學習者分享他們的舞蹈動作，或請全班一起設計幾個不同的舞蹈動作。
 - 與機器人共舞！



學習程式設計 1



玩 MeeBot 學編舞

輔導員所需材料：

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

學習者所需材料：

- 《走到邊緣再返回》學習單
- 鉛筆
- iPad 裝置 (選用)
- 額外的紙張 (選用)



[下載學習程式設計的學習單](#)



變數



概覽

第 1 課：沉下去或浮起來

- 探索：進行更新變數的討論
- 發現：沉下去或浮起來活動
- 邊玩邊學：持續追蹤和遊戲範例

第 2 課：文字遊戲

- 探索：討論問題的答案類型
- 發現：文字遊戲活動
- 邊玩邊學：地板闖關遊戲

第 3 課：關於我的一切

- 探索：討論使用列表回答問題
- 發現：關於我的一切活動
- 邊玩邊學：使用迴圈

學習者的收穫

- 將變數名稱與指定的值建立關聯
- 改變指定給變數的值
- 了解可以指定給變數的不同 Swift 類型，包含真/偽 (布林值)、數字 (整數)、文字 (字串)、顏色 (顏色字面值) 和影像 (影像字面值)
- 為指示和程式碼進行測試與除錯

詞彙

- **Variable (變數)**：用來儲存值的具名容器，而且可以改變
- **Data (資料)**：資訊
- **Boolean (布林值)**：一種類型，只有 `true` 或 `false` 這兩個可能的值

標準

1A-AP-09、1B-AP-09、1B-AP-10、1B-AP-16 >

探索

目標：透過計算物件和更新變數的數字，探索變數的概念。

輔導員所需材料：

- 白板
- 麥克筆
- 板擦
- 容器
- 五枝鉛筆 (或任何五個相同的物件)

指引：

1. 在白板上寫上一個變數語句來持續追蹤你的物件。
 - 範例：`var numberOfPencils = 0`
2. 拿起一個空的容器，並告訴學習者容器代表變數：`numberOfPencils`。
3. 將一枝鉛筆放入容器，接著詢問學習者現在變數是多少。當他們答對時，擦去數字 0 並寫上 1。
4. 繼續此動作，直到你將全部鉛筆都放入容器中，並且白板上寫著 `var numberOfPencils = 5`。
5. 然後開始將鉛筆從容器拿出，並同時更新變數。

學習重點：幫助學習者了解，變數會儲存一些資訊。在此範例中，這個資訊就是數字，而這個數字代表容器中鉛筆的數量。

發現

目標：請學習者對找到的物品做實驗，看看物品會沉下去還是浮起來，並使用影像 (影像字面值) 和真/偽 (布林值) 將資料記錄下來。

學習者所需材料：

- iPad 裝置
- Keynote app
- 《沉下去或浮起來》學習單
- 一桶水
- 要實驗的數個物件

指引：

1. 將學習者分成小組。
2. 請學習者收集多種不同的物品進行測試。
3. 請學習者對每項物品進行下列事項：
 - 拍攝照片並加入學習單。
 - 在水中測試物品。
 - 在學習單上圈出 `true` 或 `false` 來記錄結果。



[下載《沉下去或浮起來》學習單](#)

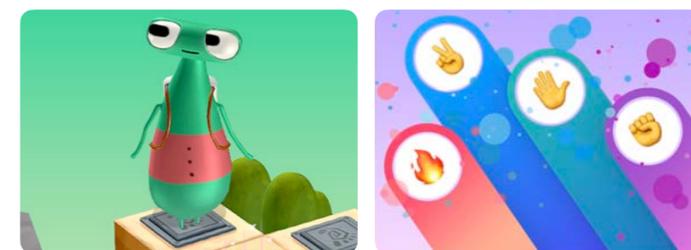
邊玩邊學

目標：學習者將能在兩種不同的程式設計情境中，建立和更新變數。

指引：

1. 將「學習程式設計 2」的 Playground 投影到螢幕上。前往「變數」章節。
2. 簡介：
 - 全班一起閱讀章節內容，若有疑問隨時暫停並討論。
3. 持續追蹤：
 - 請學習者嘗試以不同的方式，引導 Hopper 從開始箭頭處移動到有寶石的地方並收集寶石。請大家將指令記錄在學習單或另一張紙上。
 - 全班一起腦力激盪，在 Swift Playgrounds app 中編寫程式碼來完成關卡。點一下「執行我的程式碼」。
 - 嘗試多個不同的構想。
 - 與 Hopper 一起慶祝！
4. 離開「學習程式設計 2」並前往「剪刀、石頭、布」Playground 的最後一頁：「遊戲範例」。(此 Playground 頁面沒有相對應的學習單。)
5. 遊戲範例：
 - 在進行任何更動前，請點一下「執行我的程式碼」來玩這個遊戲。
 - 請與小組一起決定要自訂此遊戲的哪些部分。你可以做一些有趣的更動，包含 `game.roundsToWin`、`game.challenger.emoji`、`game.addOpponent` 和 `game.roundPrize`。
 - 再多玩幾次，每次都做些不同的更動。

延伸活動：許多變數都已建立於 Game.swift 檔案。如果學習者想知道為什麼有些變數前面沒有 `var`，請打開 Game.swift 檔案並指出建立遊戲屬性的位置。



學習程式設計 2

剪刀、石頭、布

輔導員所需材料：

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

學習者所需材料：

- 《持續追蹤》學習單
- 鉛筆
- 額外的紙張 (選用)



[下載學習程式設計的學習單](#)

探索

目標：探索現實生活中多種不同的答案類型，並將這些答案類型和多種不同的 Swift 類型建立關聯，包含是/否或真/偽 (布林值)、數字 (整數)、文字 (字串)、顏色 (顏色字面值) 和影像 (影像字面值)。

輔導員所需材料：

- 白板
- 麥克筆

討論：全班一起腦力激盪，列出會得出不同答案類型的問題並寫在白板上。

範例：

- 你的眼睛是什麼顏色？—> 顏色
- 你有養寵物嗎？—> 是/否
- 你有兄弟姊妹嗎？—> 是/否
- 你今年幾歲了？—> 數字
- 你叫什麼名字？—> 文字

學習重點：解釋變數也擁有不同類型，包含數字、文字、顏色、影像和「是」或「否」這樣的答案。即使你將變數更新為新的變數，也必須根據建立變數的方式保持其類型一致。例如：`var myAge = 8` 可以改成 9，但不可以改成「nine」。

發現

目標：學習者將能填入正確答案類型，以完成文字遊戲。

學習者所需材料：

- 《文字遊戲》學習單
- 鉛筆
- 彩色鉛筆

指引：

請學習者分小組進行一或多個文字遊戲。每組最好要有至少一位已識字成員或協助者。若所有學習者還不識字，可以全班一起進行幾個遊戲。

延伸活動：如果學習者具備足夠能力，請他們設計文字遊戲，讓夥伴填入內容。鼓勵他們在空白處填入數字、文字、顏色、影像和「是」或「否」這樣的答案。



[下載《文字遊戲》學習單](#)

邊玩邊學

目標：學習者將能引導 Byte 收集多個寶石，將每個寶石加入容器中，並且更新變數。

準備：將學習者分成三人一組。使用油漆用遮蔽膠帶，在地板上為每組建立 4 × 4 的網格。

指引：

1. 分發材料，接著將學習者分成三人一組。
2. 閱讀各角色卡，接著將角色指派給每個組員以進行第一場遊戲。
3. 請學習者從設計師角色開始來玩這個遊戲。
4. 玩三次，每次輪流使用不同的角色卡。

角色：

- 設計師：在網格上放置多個寶石和開始箭頭。
- 程式設計師：在夥伴的協助下，將指令卡放置於網格的上方或旁邊，以引導 Byte 移動到有寶石的地方並收集寶石。
- 測試人員：讓 Byte 從箭頭處開始，照著指令引導 Byte 在網格上移動，每當收集到寶石時便將寶石加入容器中。如果成功收集到全部寶石，請更新容器上的 `numberOfGems` 變數，恭喜你！如果沒有，請與小組一起修正程式碼。

替代活動：

如果學習者是與你進行一對一課程或在家自學，則可以自行下載 Keynote 替代活動來進行這個遊戲。

輔導員所需材料：

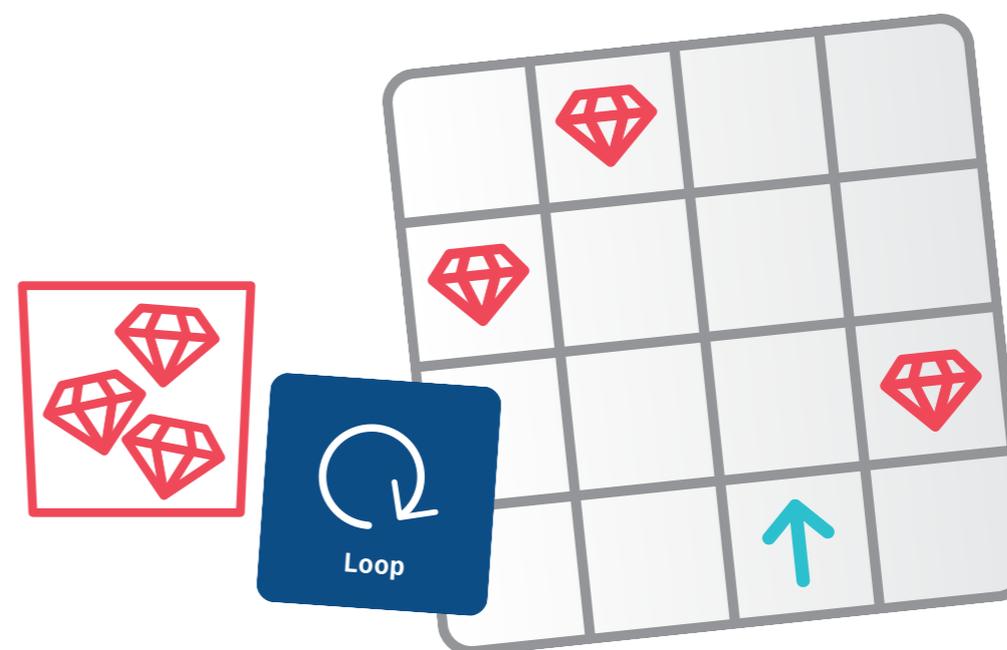
- 油漆用遮蔽膠帶

學習者所需材料：

- 角色卡
- 指令卡：`moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `Loop`
- 寶石
- Byte
- 箭頭
- 標示以下標籤的容器：
`var numberOfGems = _____`
- 筆

↓ [下載材料](#)

↓ [下載替代活動](#)



探索

目標：探索如何使用列表 (也稱為陣列) 來建立變數。

討論：試想一下，如果學習單要求學習者填入兄弟姊妹的名字，但是他們卻有多位兄弟姊妹，則會發生什麼事情呢？聽聽班上同學怎麼說。如果他們的建議是建立一個列表，告訴他們這就是身為程式設計者才會做的事！當有個變數的答案不止一個時，學習者就應該建立列表。

請學習者提出一些可能擁有多個答案的問題。

範例：

- 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
- 最喜歡的顏色 —> 、、、、
- 最喜歡的動物 —> 、、、

學習重點：學習者在程式碼中建立的列表，和句子中的列表是一樣的概念。

發現

目標：學習者將能填入變數，以描述關於自己和夥伴的事情。學習者可能有機會使用變數類型之一的陣列。

學習者所需材料：

- 《關於我的一切》和《關於你的一切》學習單
- 鉛筆
- 彩色鉛筆

指引：

1. 請學習者完成「關於我的一切」學習單。
 - 如果學習者擁有不只一個兄弟姊妹或寵物，請他們建立項目列表並以逗號分隔。
2. 將學習者分為兩人一組，一起完成《關於你的一切》學習單。

替代活動：學習者可以使用自己的 iPad 和 Keynote 來完成學習單，並為圖片答案拍攝照片，還有使用格式選項為顏色字面值上色。



[下載關於一切學習單](#)

邊玩邊學

目標：學習者將能識別程式碼中的變數，並探索如何搭配迴圈使用陣列。

指引：

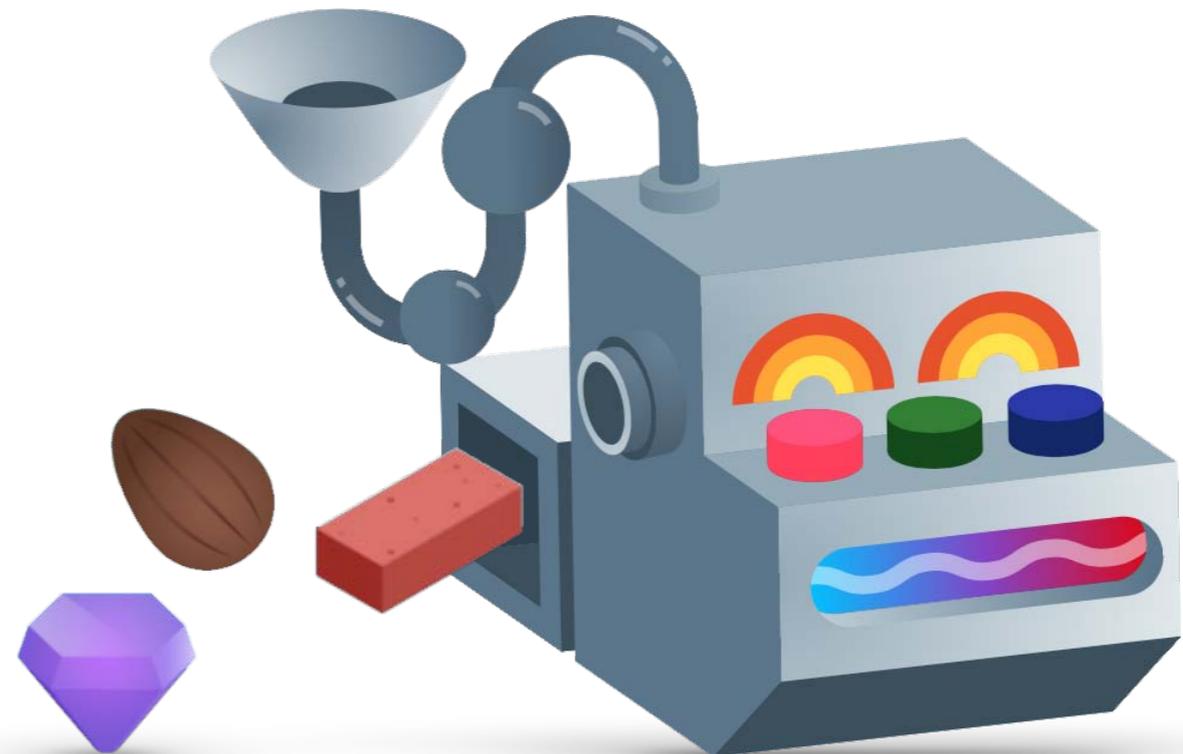
1. 將「程式碼機器」的 Playground 投影到螢幕上。
2. 簡介：
 - 全班一起閱讀章節內容，若有疑問隨時暫停並討論。
 - 選擇性：播放前兩頁，也就是「探索機器」和「使用顏色結合」。
3. 使用迴圈：
 - 在此頁面上，學習者會將自己的迴圈知識與變數結合。
 - 看看學習者是否能識別程式碼中使用陣列的變數。
 - 點一下「執行我的程式碼」，查看機器製作出什麼東西。
 - 前往指示中的第二步驟，接著更新程式碼，使其包含第二個變數、物品及嵌套迴圈。
 - 再次點一下「執行我的程式碼」，查看機器製作出什麼東西。
 - 注意：在進行活動之前，請和學習者一起嘗試進行這一頁。



程式碼機器

輔導員所需材料：

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



App 設計



探索

目標：在各式各樣的裝置上探索熟悉的 app。

說明：對學習者在家或在校使用的 iPad app 展開討論。接著討論學習者、學習者的家長或監護人在家使用的 app。

學習重點：加強概念，表示 app 並不只手機有，也存在於手錶、平板電腦、電腦，甚至是電視上。

延伸活動：深入探討一些 app 範例，請學習者思考 app 的設計對象、功用以及設計原因。

範例：

- App: Swift Playgrounds
- 設計對象: 想要了解 Swift 的人
- 功用: 以關卡和課程的方式，幫助學習者學習程式設計
- 設計原因: 教導略懂程式設計或毫無經驗的學習者如何編寫程式碼

發現

目標：透過分析熟悉的 app，幫助學習者做好準備，以設計出自己專屬的 app。

學習者所需材料：

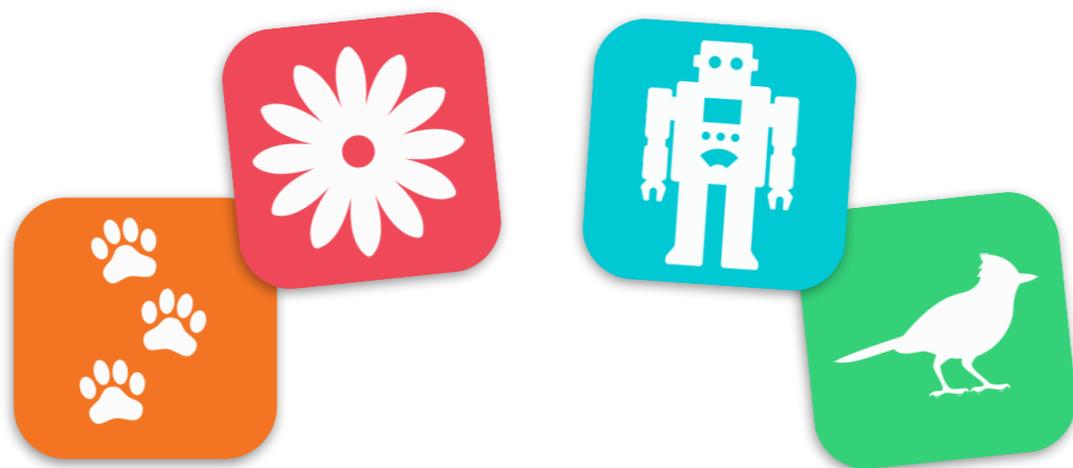
- iPad 裝置
- 《App 是什麼?》學習單
- 鉛筆
- 彩色筆或鉛筆

指引：

1. 將學習者分成小組，或讓他們各自獨立作業。
2. 請學習者選擇一個 iPad app。
3. 請學習者使用「App 是什麼?」學習單來引導他們的 app 探索過程。
4. 請學習者分享他們的發現，與小組或是夥伴分享皆可。

給輔導員的提示：當學習者年齡越小，你須要提供更多幫助來完成這份學習單。針對幼兒園至一年級的學童，請考慮一組選兩個或三個 app。

↓ [下載《App 是什麼?》學習單](#)



邊玩邊學

目標：學習者設計自己專屬的 app！

學習者所需材料：

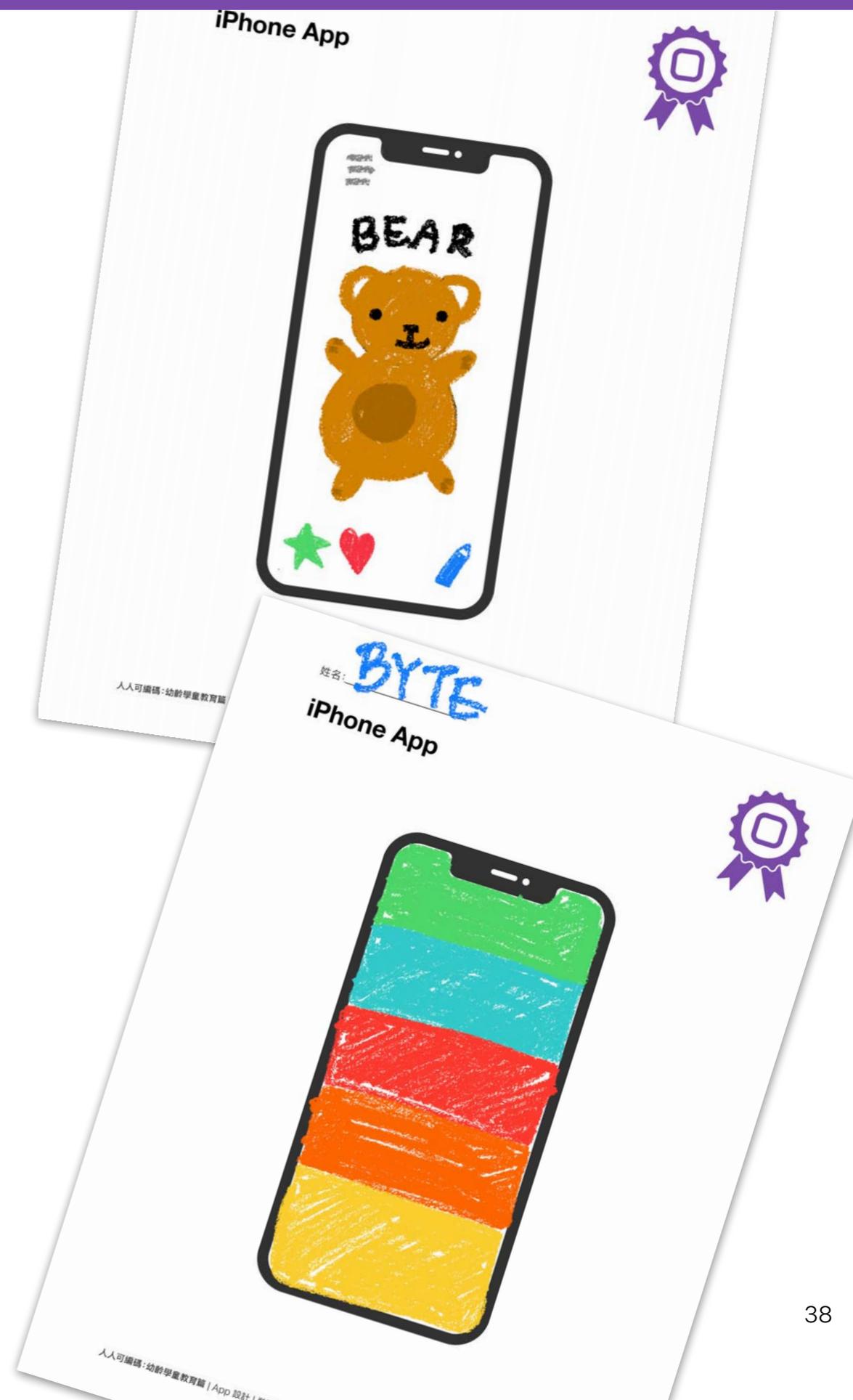
- 《我的 App 設計》學習單
- 裝置樣板
- 額外的紙張
- 鉛筆
- 彩色筆或鉛筆

指引：

1. 將學習者分成小組，或讓他們各自獨立作業。
2. 為了在設計 app 的起始階段引導學習者，請帶著他們逐步進行《我的 App 設計》學習單。
3. 請學習者利用額外的紙張或裝置樣板，製作他們 app 頁面的原型。
4. 指導學習者利用裝置樣板來建立他們 app 原型的最終版本。
5. 邀請學習者以個人或小組的形式，向大家展示自己的構想。

↓ [下載《我的 App 設計》學習單](#)

↓ [下載裝置樣板](#)



輔導員資源



詞彙表

- **Boolean (布林值)**: 一種類型, 只有 true 或 false 這兩個可能的值
- **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 (資料)**: 資訊
- **Debug (除錯)**: 找到並修正程式碼中的錯誤
- **Function (函數)**: 一組具名的指令, 能在需要時執行
- **Loop (迴圈)**: 會重複執行一定次數的程式碼區塊
- **Modify (修改)**: 做出改變
- **Sequence (序列)**: 事情發生的順序
- **Step (步驟)**: 一連串程序中的一個動作
- **Toggle (切換)**: 打開或關閉
- **Variable (變數)**: 用來儲存值的具名容器, 而且可以改變

CSTA 標準 >

1A-AP

- 1A-AP-08: 透過建立和遵循演算法, 為日常工作流程建立模型 (多組逐步指示) 來完成任務。
- 1A-AP-09: 使用數字或其他符號, 為程式儲存和操縱資料的方式建立模型來呈現資訊。
- 1A-AP-10: 使用序列和簡單的迴圈來開發程式, 以表達構想或解決問題。
- 1A-AP-11: 將解決問題所需步驟解構 (分解) 成精準的指示序列。
- 1A-AP-12: 制定計畫, 說明程式的事件、目標和預期結果的序列。
- 1A-AP-14: 為包含序列和簡單迴圈的演算法或程式除錯 (識別和修正錯誤)。

1A-CS

- 1A-CS-01: 選擇及操作適當的軟體以執行各種不同的任務, 並了解使用者對自己使用的科技產品擁有不同的需求和偏好。

1B-AP

- 1B-AP-09: 製作出利用變數來儲存及修改資料的程式。
- 1B-AP-10: 製作出包含序列、事件、迴圈和條件的程式。
- 1B-AP-16: 在輔導員指導下扮演各種角色, 與夥伴合作進程式開發的設計、實行與審查階段。

範例解答

以下幾頁為 Swift Playgrounds 的每個關卡提供一個可行的解決方案，但是過關的方式不只一種。鼓勵學習者嘗試用任何他們可以想到的方式來引導 Byte 或其他角色。

肯定學習者想出來的所有編寫程式碼的方式，以及可能擁有的任何目標。有些學習者可能會想一邊收集寶石，一邊探索關卡的各個角落，而有些人則選擇在路上盡可能多次往返來收集寶石。別忘了，程式設計應該充滿樂趣！



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()
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

本頁面沒有提供解答範例，
因為這個遊戲可以完全隨你
愛怎麼玩就怎麼玩！



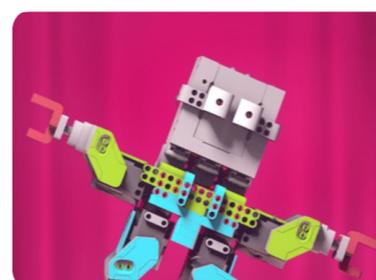
程式碼機器

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

基礎舞步

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

舞步循環

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

更進一步



《人人可編碼：解謎闖關》教師指南和《人人可編碼：探險闖關》教師指南教師指南專為支援教育工作者而設計，讓他們帶著自信教導程式碼，並透過與現實世界互動、溝通、團隊合作、批判性思考和個人化學習，來深化學生的學習。教師指南同時也為不同的課程活動提供評量建議和提示。[下載《人人可編碼：解謎闖關》教師指南和《人人可編碼：探險闖關》教師指南](#)>

Apple Teacher 計畫

Apple Teacher 是一套免額外付費、可自訂進度的專業學習計畫，提供可無限存取的教材與內容，協助你在教育工作中善用 Apple 科技產品的力量。[進一步了解](#)>

Apple 專業學習

Apple 專業學習專家帶領引人入勝的實作學習體驗，幫助教育工作者發展創新教學實務，吸引學生更投入地學習。如需進一步了解，請傳送電子郵件至 aplshk@apple.com。



人人可編碼課程的 Swift 編碼社團

「Swift 編碼社團」是在課後輔導、夏令營或其他非正式的學習環境中介紹程式碼的絕佳方法。「Swift 編碼社團」的單元設計對於初次接觸程式編寫的新手，以及較有經驗的學生都很適合。[下載 Swift 編碼社團工具組](#)>



程式編寫快速入門

這本指南提供 10 種有趣的程式設計活動，適合 10 歲以上的學生使用。學生無論在學校或家裡，都可透過 iPad 與 Mac 上的免額外付費 Swift Playgrounds app 學習編寫程式碼。[下載《程式編寫快速入門》](#)>



App 設計日誌

透過《App 設計日誌》，學生可以應用 app 設計程序，解決他們在學校或社群中遇到的問題。這本日誌會提示學生針對自己的 app 構想進行腦力激盪、規劃、原型製作與評估，最後以 app 原型的提案簡報畫上完美句點。[下載《App 設計日誌》](#)>



© 2022 Apple Inc. 保留一切權利。Apple、蘋果、Apple 標誌、Apple Watch、iPad、iPhone、Keynote、Mac、Pages、Swift、Swift 標誌和 Swift Playgrounds 是 Apple Inc. 在美國及其他國家或地區的註冊商標。App Store 和「人人可編碼」是 Apple Inc. 在美國及其他國家或地區註冊的服務商標。此處提及的其他產品和公司名稱可能為其所屬公司的商標。2022 年 11 月