人人能编程
学前班至低年级
教学篇

教师指南

目录

《人人能编程：学前班至低年级教学篇》旨在帮助教育工作者和家长带领孩子从低年级就开始认识编程，此时孩子刚刚开始形成计算思维技能。在这些课程中，幼儿园到三年级的学习者能够通过一系列探索、发现和练习活动，打好核心编程概念的基础。

指导设计

本教师指南由四个模块和一个最终 app 设计项目组成。每一个模块都包括三节课，每节课重点介绍一个编程相关概念。每节课都涵盖三项活动：探索、发现和练习。这些活动可以细分为多个阶段或分几天完成。

第 1 天：讨论和实际操作学习

探索

介绍编程概念并加以探讨

发现

通过创意活动熟悉编程概念

~25 分钟

练习

- 在 Swift Playgrounds app 中与 Byte 一起编程
- 在配套工作表和 Keynote 讲演活动中练习编程
- 借助不插电的拼图编程游戏带领 Byte 走进现实世界

~25 分钟

第 2 天：将学习与代码关联起来

范围与顺序

本教师指南包括四个模块，适用于幼儿园至三年级教学。四个模块可按任意顺序教授。其中，“App 设计”模块可在一年中的任何时候教授，随着学习者对代码和 app 的了解不断加深，可反复多次教授该模块。

示例：

年级	模块	最终项目	总时长 (近似值)
幼儿园	命令	App 设计	4 小时
一年级	函数	App 设计	4 小时
二年级	循环	App 设计	4 小时
三年级	变量	App 设计	4 小时

继续学习

对于四至八年级的教学，《人人能编程：解谜闯关》和《人人能编程：探险闯关》课程以及《App 设计日志》与《App 展示活动指南》可提供 90 多小时的教学时间。请查阅《人人能编程课程指南》了解更多信息。

学习者作品集 (可选)

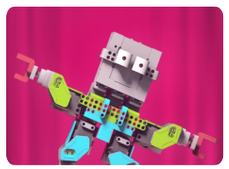
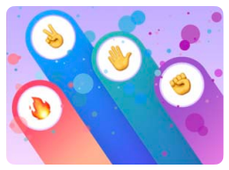
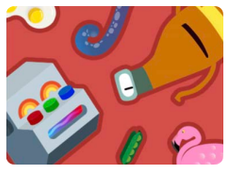
收集这些模块活动中完成的作品，与学习者一同创建作品集。

模块	课程	建议工作
命令	例行事项	• “发出命令”工作表 • “添加新命令”工作表
	故事的顺序	• “故事的顺序”情节图片 • “故事的顺序”小组照片
	舞蹈动作	• 舞蹈动作卡 • “舞蹈动作”视频 (可选)
函数	纸宝石	• 纸宝石形状 • “组合新行为”工作表 • “创建新函数”工作表
	合唱会	• 合唱演唱会视频或编写的函数
	我的静心函数	• “我的静心函数”图片或视频 • “收集、切换、重复”工作表
循环	重复花瓣	• 重复花瓣 • “使用循环”工作表 • “循环每一侧”工作表
	障碍路线	• 障碍路线视频或图片 (可选)
	集体鼓乐	• “行至边缘再返回”工作表 • 集体鼓乐视频或图片 (可选)
变量	沉没或漂浮	• 沉没或漂浮 • “跟踪记录”工作表
	文字游戏	• 文字游戏
	自我介绍	• 自我介绍 • 你的简介
App 设计		• “App 是什么?” • 我的 App 设计 • “App 设计”原型

开始在 iPad 或 Mac 上使用 Swift Playgrounds

开始上课前，请确保已将 Swift Playgrounds、Pages 文稿和 Keynote 讲演下载到设备上。

本教师指南的各个模块分别搭配使用了不同的 Playground。下面列出了每个模块需要用到的 Playground：

模块	Playground			如何在 Swift Playgrounds 中下载
命令	 学习编程 1	 玩 MeeBot 学跳舞		要订阅 MeeBot Playground Feed，请滚动到“更多 Playground”屏幕底部，然后轻点“输入订阅 URL”。然后输入：ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json。
函数	 学习编程 1			
循环	 学习编程 1	 玩 MeeBot 学跳舞		要订阅 MeeBot Playground Feed，请滚动到“更多 Playground”屏幕底部，然后轻点“输入订阅 URL”。然后输入：ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json。
变量	 学习编程 2	 石头剪刀布	 编程机器	“石头剪刀布”和“编程机器”可以在“更多 Playground”屏幕的“图书”部分中找到。
App 设计				

请访问 App Store 了解 Swift Playgrounds 的最低要求。请访问 Apple 支持获取 Swift Playgrounds 相关帮助。

协导员提示

为了帮助学习者在课程中收获最大化，请尽量按照以下提示进行操作。

“探索和发现”活动：

- 在编写或展示代码时，应简化所有语法或特殊事例，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 =`

练习活动：

- 为了让 Swift Playground app 对低龄学习者更加友好，请按课程计划中的说明进行操作。其中包含以下内容：
  - 整个小组一同朗读说明
  - 向学习者提供简化的指令，以帮助他们完成工作表，并提出自己的解决方案
  - 使用协导员的 iPad 或 Mac 来打通 app 中的关卡
- `let` 与 `var`：本指南不涉及 `let` 关键字。为避免在 Swift Playgrounds 中造成混淆，在向学习者展示页面前，请将所有 `let` 关键字改为 `var`。在我们建议使用的 Playground 中，这两个关键字可互换使用。
  - `let` = 不会改变的变量
  - `var` = 会改变的变量

拓展：

- 拓展在地面上开展的“练习”活动，纳入算术、识字、常见字练习、拼写等。尝试在“函数”模块中进行地面“练习”活动，激发灵感。
- 请学习者自行为 `twirl()` 或 `jump()` 等命令制作卡片，使地面“练习”活动因人而异。

介绍页面

Playground 页面

命令

概述

第 1 课：例行事项

- 探索：就烘焙与命令之间的关联展开讨论
- 发现：“例行事项”活动
- 练习：发出命令和添加新命令

第 2 课：故事的顺序

- 探索：就故事情节顺序与命令之间的关联展开讨论
- 发现：“故事的顺序”活动
- 练习：拼图游戏

第 3 课：舞蹈动作

- 探索：就舞蹈动作与命令之间的关联展开讨论
- 发现：“舞蹈动作”活动
- 练习：“你好，MeeBot”和“基础舞步”

学习者将习得以下技能

- 以日常生活为例描述分步指令
- 按照正确顺序排列指令，使其发挥应有的作用
- 测试和调试相关指令和代码

词汇

- **Sequence (序列)**：事情发生的顺序
- **Step (步骤)**：一个大规模流程中的一项操作
- **Modify (修改)**：更改
- **Command (命令)**：能够让应用程序执行特定操作的代码
- **Bug (错误)**：代码中的错误
- **Debug (调试)**：找出代码中的错误并加以修复

标准

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

探索

目标：将命令与烘焙布朗尼蛋糕关联起来，以此介绍命令的概念。

讨论：

- 他们会按照布朗尼蛋糕食谱进行操作吗？
- 他们会按照布朗尼蛋糕食谱的步骤顺序进行操作吗？

要点：食谱中的步骤或指令相当于代码中的命令。让学习者自己构思出一些命令。

发现

目标：确定某个例行事项的分步指令，然后为该例行事项的流程建模。

材料：洗手卡

做法：

1. 对“洗手”卡进行洗牌，然后将这些卡片摆在桌子上或贴在白板上。打乱卡片的顺序。
2. 询问学习者你排列的洗手顺序是否有误。
3. 请学习者每次移动一张卡片，使其移至正确位置，以此纠正指令错误。

替代方案：

让学习者两两搭档或分小组行动，然后给每组发放一套卡片。

拓展：

让学习者为自己每天都要做的一件事写下分步指令，并画出具体步骤。

下载“洗手”卡

练习

目标：学习者能够在 Swift Playgrounds app 的“学习编程 1”中以正确顺序添加命令，收集自己的第一颗宝石。

做法：

1. 将“学习编程 1”Playground 中“命令”一章的介绍页面投影到屏幕上。
2. 介绍：
   - 全班一同朗读页面内容，必要时可暂停并提问。
3. 发出命令：
   - 复习将 Byte 移动到宝石处所需使用的两个命令，也就是 `moveForward()` 和 `collectGem()`。
   - 让学习者尝试通过不同方法引导 Byte 从开始箭头处移动到宝石处并加以收集。学习者可以将命令记录在工作表或另一张纸上。
   - 收集全班学习者的想法，然后在 Swift Playgrounds app 上编写代码，完成闯关。点按或轻点“运行我的代码”。
   - 试一试其他思路。
   - 恭喜，Byte 闯关成功！

拓展：

如果学习者已经准备完毕，请转至下一页，即“添加新命令”。学习者将在这一页学习使用新命令 `turnLeft()`。

学习编程 1

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “学习编程 1”Playground
- 投影仪或显示器

学习者所需材料：

- “发出命令”和“添加新命令”工作表
- 铅笔
- 其他纸张 (可选)

下载“学习编程”工作表

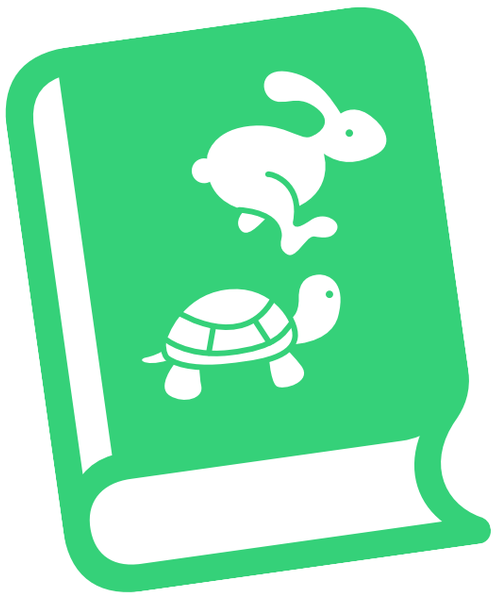
探索

目标：探索图书如何按照一定的顺序 (从开端到发展再到结局) 合理推进情节。

讨论：

- 询问学习者图书是否符合一定的顺序。
- 如果图书的开端、发展和结局顺序被打乱，会怎样？
- 探索几个示例。

要点：与代码关联起来，强调以正确顺序编写编程命令的重要性，就像故事情节的顺序一样重要。



发现

目标：学习者为故事中的各个情节绘制图片后，能够按照正确的顺序排列图片，并准确地复述故事。

协导员所需材料：

- 白板
- 马克笔

学习者所需材料：

- 纸
- 马克笔或彩色铅笔
- 替代方案：iPad 设备和绘图 app

做法：

1. 朗读一个学习者们都知道的故事。全班一同确定这个故事的主要情节。最好能够找出四到六个情节。
2. 将学习者分组，每组人数应当与情节数量相同，例如，如果有四个情节，则每个小组应当有四个学习者。
3. 让小组中的每位学习者分别画出一个情节。
4. 各小组轮流站在教室前，每位小组成员分别拿着自己的情节图，打乱顺序站开。
5. 观众重新排列图片顺序，一次移动一张图片。
6. 当小组成员的站位调整为正确顺序后，拍下此时的照片。

拓展或替代方案：

让各组学习者选择不同的故事，先一同确定情节，然后再绘制图片。

练习

目标：学习者能够利用方向命令引导 Byte 在实际的网格中移动并到达宝石处。

准备：学习者三人一组完成任务。使用 Painter 胶带在地面上为每组制作一个 4×4 的网格。

做法：

1. 分发材料，然后将学习者分为三人一组。
2. 阅读每个角色的介绍，然后为组内每位成员分配角色，为第一个游戏做准备。
3. 让学习者玩游戏，先从设计师角色开始。
4. 玩三次，每次轮换一遍角色卡。

角色：

- 设计师：将宝石和开始箭头放入网格。
- 程序员：在同伴的帮助下，将命令卡放在网格中或网格旁，引导 Byte 移动到宝石处并加以收集。
- 测试员：开始时让 Byte 位于箭头上，然后按照命令卡让 Byte 在网格中移动。如果能够收集到宝石，说明编程成功，恭喜你！如果没有收集到宝石，请一同调试或修复代码。

协导员所需材料：

- Painter 胶带

学习者所需材料：

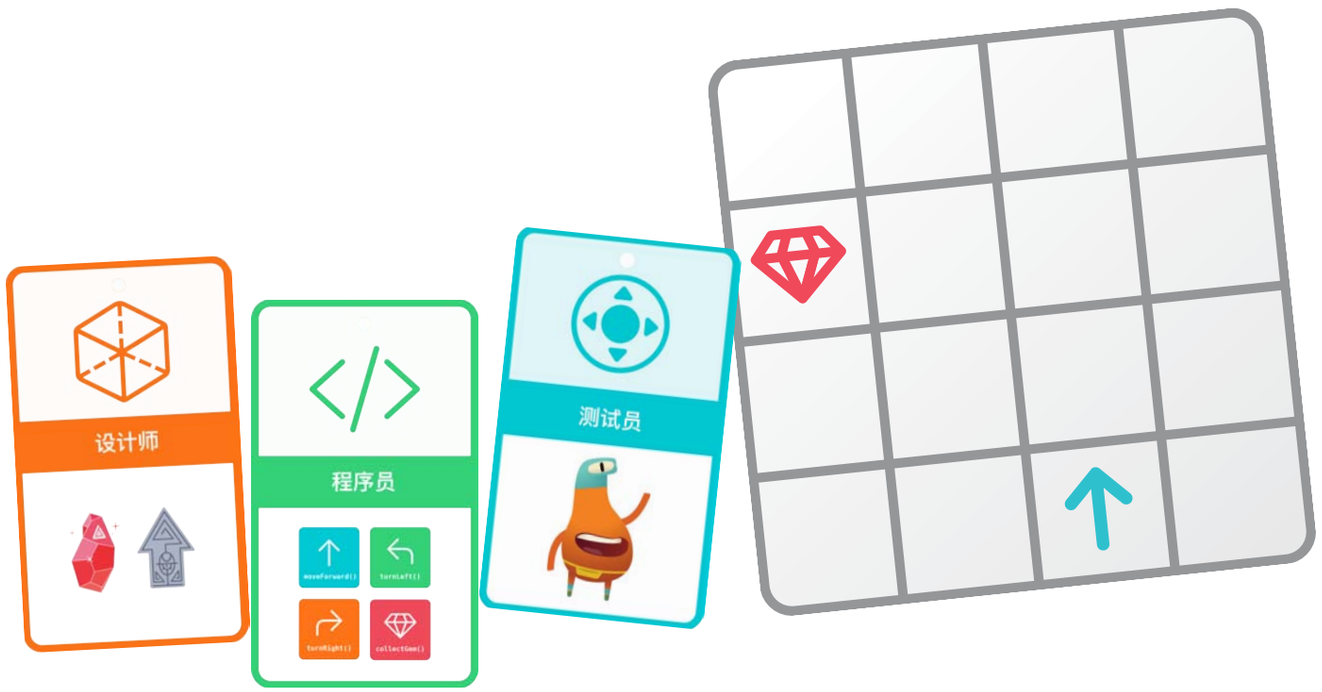
- 角色卡
- 命令卡：`moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()`
- 宝石
- Byte
- 箭头

下载材料

下载替代活动

替代方案：

如果你与学习者一对一教学，或者学习者在家学习，则可使用可下载的 Keynote 讲演替代活动自行玩这个游戏。



探索

目标：探索“编程也可以创意十足”这一理念！

讨论：

- 询问学习者是否学过舞蹈。
- 舞蹈动作是否遵循一定的舞步顺序？
- 他们是怎么知道下一个舞蹈动作是什么的？
- 舞蹈动作有名字吗？
- 学习者是否会在一支舞蹈的不同时间节点使用相同的舞蹈动作，或者在不同舞蹈中使用相同的舞蹈动作？

要点：帮助学习者联想到编程也可以创意十足，与此同时，和编舞一样，编程人员也可以编写新的命令，然后以各种有趣的方式将新命令糅合在一起。

发现

目标：编排一段简短的舞蹈，同时使用卡片来表示舞蹈动作。每一张舞蹈动作卡就好比是“学习编程” Playground 中的一个命令。

学习者所需材料：

- iPad 设备
- Keynote 讲演 app
- “相机”app
- 跳舞空间

做法：

1. 让学习者两人或多人一组，一同编排一段简短的舞蹈。
2. 学习者确定舞蹈后，需要为不同的舞蹈动作制作卡片。学习者应在每张卡片上绘制图样并注明对应的舞蹈动作名称，尽量做到有新意、有趣。
3. 各小组分别跳起自己的舞蹈，然后全班一起来开个舞会！

替代方案：

学习者可以使用下方可下载的舞蹈动作卡来编排舞蹈，也可以将这些卡片作为参照，自行制作舞蹈动作卡。

拓展：

学习者制作舞蹈视频，然后向小组成员展示。

下载舞蹈动作卡

练习

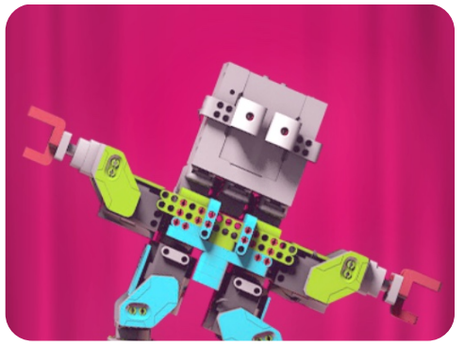
目标：创建一个步骤序列来教 MeeBot 机器人一段新的舞蹈。

做法：

1. 将“玩 MeeBot 学跳舞”Playground 投影到屏幕上。如果你还没有“玩 MeeBot 学跳舞” Playground，请订阅此 Playground。
2. 介绍：
   - 全班一同朗读页面内容，必要时可暂停并提问。
3. 你好，MeeBot：
   - 点按或轻点“运行我的代码”，然后观看机器人跳舞。
4. 基础舞步：
   - 让学习者以小组为单位，两人一组或独自使用自己的 iPad 从建议列表中选择八个命令，然后观看机器人如何跳舞。
   - 让学习者分享其舞蹈，或全班一同编排一些其他的舞蹈。
   - 和机器人一起跳起来吧！

拓展：

- 前往下一页 (即“舞步组合”)，让学习者在 `myDanceRoutine()` 函数中添加舞蹈动作，命令数量可多可少，视需求而定。



玩 MeeBot 学跳舞

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “玩 MeeBot 学跳舞”Playground
- 投影仪或显示器

学习者所需材料：

- iPad 设备 (可选)

函数

概述

第 1 课：纸宝石

- 探索：就分步指令展开讨论
- 发现：“纸宝石”活动
- 练习：“组合新行为”和“创建新函数”

第 2 课：合唱会

- 探索：就如何命名函数展开讨论
- 发现：“合唱会”活动
- 练习：拼图游戏

第 3 课：我的静心函数

- 探索：就通过多种方式解决问题展开讨论
- 发现：“我的静心函数”活动
- 练习：收集、切换、重复

学习者将习得以下技能

- 将庞大的问题或任务分解成若干小步骤
- 创建一系列步骤来解决某个问题或完成某项任务
- 为函数命名
- 测试和调试代码

词汇

- **Function (函数)**：一组可视需要随时运行的具名命令
- **Toggle (切换)**：打开或关闭

标准

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

探索

目标：了解对一系列命令进行封装的概念及封装函数的命名。

讨论：选出一个例行事项供全班讨论。让学习者说出例行事项的名称，并介绍该例行事项包括哪些步骤。

示例：睡前例行事项
- 第 1 步：刷牙
- 第 2 步：上洗手间
- 第 3 步：看书
- 第 4 步：互道晚安
- 第 5 步：关灯

要点：构思一组指令并为其命名，就相当于是在创建函数。

拓展：询问学习者能否给出更详细的步骤说明。例如，刷牙的具体步骤是什么？

发现

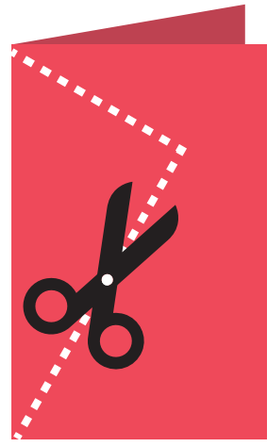
目标：学习者先根据指令制作一块纸宝石，然后写出或者画出制作另一种自选形状的指令。

学习者所需材料：
- 纸
- 剪刀
- 铅笔
- iPad 设备 (可选)

做法：

向学习者演示如何制作纸宝石：

1. 将一张纸对折。
2. 从折线的顶角斜向下画一条线，这条线的末端要在纸面中线上方 2.5 至 5 厘米处。
3. 将刚才画的线的末端与折线的底角用一条线连起来。
4. 沿着所画的线进行裁剪。
5. 从废纸上取下“宝石”并展开。



让学习者自选形状并制作出来：

1. 将学习者分成几个小组。
2. 每个小组确定一个想要制作的形状。
3. 留出时间，让学习者练习一到两次如何制作所选形状。
4. 让学习者写出或者画出制作形状的指令，并为指令命名，例如“制作圆圈”或者“字母 T”。

替代方案：

拍摄一段视频来介绍如何制作形状。

练习

目标：全体学习者一同参与，逐一分解各必要步骤，让 Byte 到达宝石处。

做法：

1. 将“学习编程 1”Playground 投影到屏幕上。导航到“学习编程 1”的“函数”一章。
2. 介绍：
   - 全班一同朗读页面内容，必要时可暂停并提问。
3. 组合新行为：
   - 复习 `moveForward()`、`turnLeft()` 和 `collectGem()` 命令。请记住，你没有 `turnRight()` 命令。
   - 让学习者尝试通过不同方法引导 Byte 从开始箭头处移动到宝石处并加以收集。学习者需要将命令记录在工作表或另一张纸上。
   - 收集全班学习者的想法，然后在 Swift Playgrounds app 上编写代码，完成闯关。点按或轻点“运行我的代码”。
   - 试一试其他思路。
   - 恭喜，Byte 闯关成功！
4. 创建新函数：
   - 根据学习者在上一个 Playground 页面 (也就是“组合新行为”页面) 学到的知识，让他们思考如何创建 `turnRight()` 函数。
   - 让学习者使用自己创建的 `turnRight()` 函数，通过不同方法引导 Byte 从开始箭头处移动到关闭的开关处并切换开关。
   - 收集全班学习者的想法，然后在 Swift Playgrounds app 上编写代码，完成闯关。点按或轻点“运行我的代码”。
   - 试一试其他思路。
   - 恭喜，Byte 闯关成功！难题迎刃而解！

学习编程 1

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “学习编程 1”Playground
- 投影仪或显示器

学习者所需材料：

- “组合新行为”和“创建新函数”工作表
- 铅笔
- 其他纸张 (可选)

下载“学习编程”工作表

探索

目标：将有关命令和函数的知识运用到歌曲上，为歌曲提供描述性的名称。

讨论：让学习者选择多首歌曲，并为每首歌曲提供一个描述性的函数名称。

示例：对于《一闪一闪小星星》这首歌，函数调用可以是 `singTwinkle()`，而 `singSong1()` 则不太合适，因为第一首歌可能会发生变化。

要点：一定要使用描述性名称来命名函数，因为这样更方便你或他人理解代码。

发现

目标：学习者将在演唱会函数中调用不同的歌曲命令，以此打造一场演唱会。

协导员所需材料：

- iPad 或 Mac
- 投影仪或显示器
- 白板
- 马克笔

做法：

1. 帮助学习者为多首歌曲创建函数名称，例如 `singHappyBirthday()`。
2. 让小组成员一同选择歌曲顺序。
3. 为一场演唱会编写一个函数定义，然后将歌曲命令填入函数。

示例：

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

替代方案：

学习者分组唱歌，每组都有自己的歌单、歌曲函数名称和歌曲顺序。然后每一组开始唱自己选择的歌曲，并为自己的演唱会录制视频。

练习

目标：学习者需要解开一个简单的方程式，将宝石放在答案对应的方格上，然后使用方向命令引导 Byte 在网格中移动。

准备：学习者三人一组完成任务。使用 Painter 胶带在地面上为每组制作一个 4×4 的网格。将开始箭头放在一个方格中，然后在剩下的所有方格中各放一个数字。

做法：

1. 分发材料，然后将学习者分为三人一组。
2. 阅读每个角色的介绍，然后为组内每位成员分配角色，为第一个游戏做准备。
3. 让学习者玩游戏，先从设计师角色开始。
4. 玩三次，每次轮换一遍角色卡。

角色：

- 设计师：掷两次骰子。在同伴的帮助下，将两次掷出的数字相加，然后将宝石放在网格中具有与所得之和对应的数字的方格上。
- 程序员：在同伴的帮助下，将命令卡放在网格中或网格旁，引导 Byte 移动到宝石处并加以收集。
- 测试员：开始时让 Byte 位于箭头上，然后按照命令卡让 Byte 在网格中移动。如果能够收集到宝石，说明编程成功，恭喜你！如果没有收集到宝石，请一同修复代码。

替代方案：

如果你与学习者一对一教学，或者学习者在家学习，则可使用可下载的 Keynote 讲演替代活动自行玩这个游戏。

协导员所需材料：

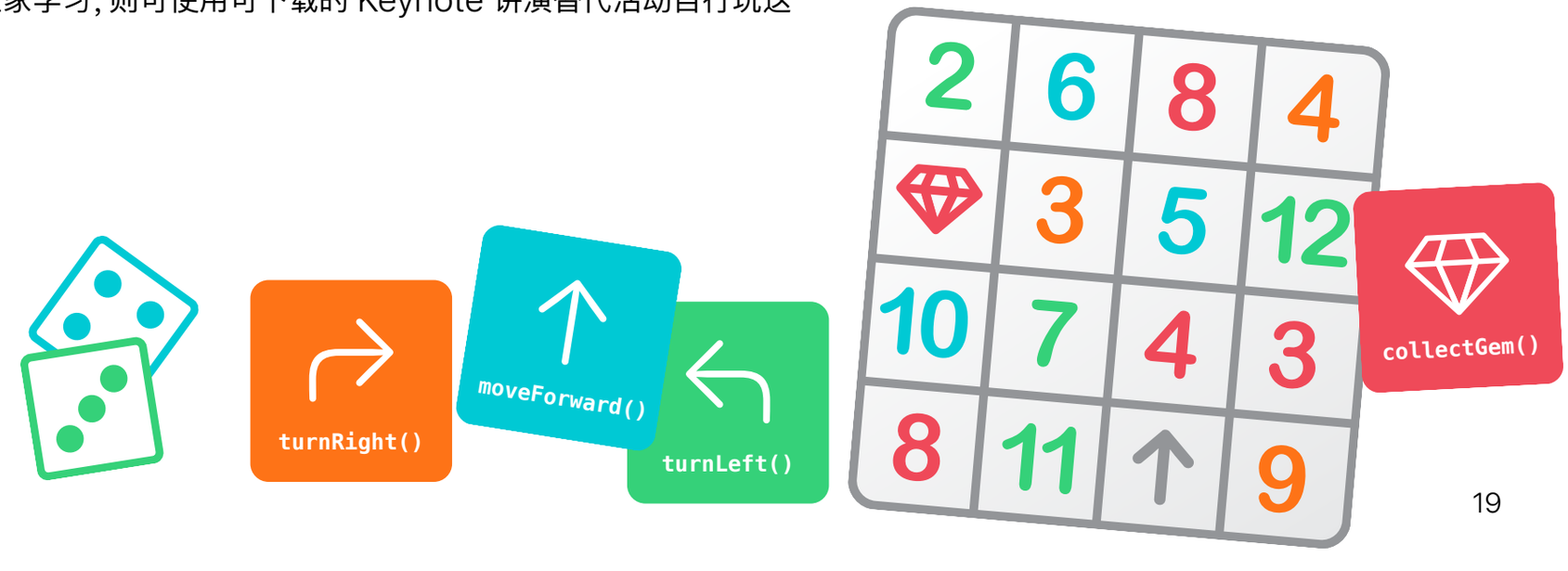
- Painter 胶带
- 为每个网格提供一组印刷体数字

学习者所需材料：

- 角色卡
- 命令卡：`moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()`
- 宝石
- Byte
- 箭头
- 两个骰子

下载材料

下载替代活动

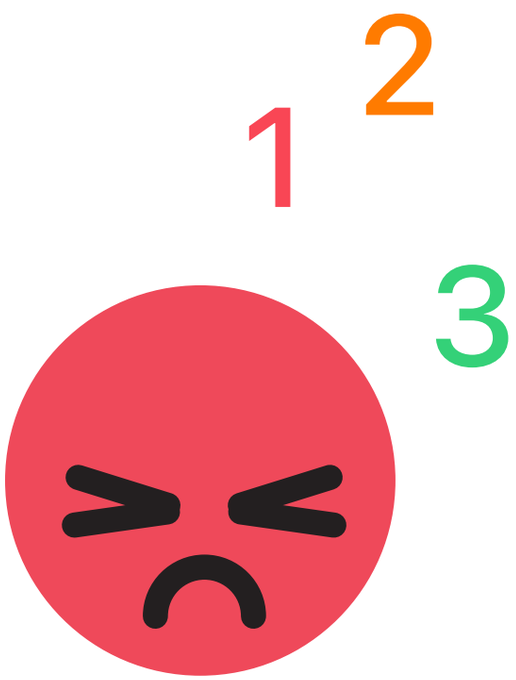


探索

目标：让学习者认识到解决问题的方法往往不止一种。

讨论：让学习者想一想曾经遇到过的问题，然后告诉大家自己是怎么解决这个问题的。询问小组成员能否使用其他方法解决这个问题。探讨多个不同的问题及其解决方案。

要点：帮助学习者找出与代码的关联，并让学习者认识到解决编程问题的方法往往不止一种。



发现

目标：学习者编写函数来展示其静心技巧，并为函数命名。

学习者所需材料：

- “我的静心函数”工作表
- 铅笔
- 彩色画笔或彩色铅笔

做法：

提示：如有可能，学习者最好独自完成这项任务。

1. 请全班学习者集思广益，多想一些能够让自己在心烦意乱时平静下来的方法，无论是在家中还是在学校时。让学习者将其静心技巧分解成若干步骤。
2. 发放“我的静心函数”工作表，然后要求学习者画出其静心技巧的步骤。
3. 让学习者为其静心技巧命名。他们可以使用驼峰式拼写法，例如 `countToTen()`，或直接使用短句，如“Count to ten”。

拓展：

不插电：让学习者在小组中或在全班面前演示其静心技巧。

使用 iPad 时：学习者制作视频来介绍其静心技巧并与全班分享。

下载“我的静心函数”工作表

练习

目标：学习者能够编写一个由多种不同类型的命令组成的函数，然后利用此函数打通关卡。

做法：

1. 将“学习编程 1”Playground 中的“收集、切换、重复”页面投影到屏幕上，指出学习者将要帮助填写的空白函数。
2. 收集、切换、重复：
   - 复习 `moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `toggleSwitch()` 命令。
   - 让学习者尝试找出关卡中的重复环节，然后发挥自己的创意，在 app 中填写函数并为函数命名。
   - 让学习者为函数创造一个符号，然后将这个符号和函数名称记录在工作表的命令键中。
   - 学习者利用这一附加的命令，尝试通过多种方式引导 Byte 收集所有宝石并切换所有开关。学习者需要将命令记录在工作表或另一张纸上。
   - 收集全班学习者的想法，然后在 Swift Playgrounds app 上编写代码，完成闯关。点按或轻点“运行我的代码”。
   - 尝试一些其他的解决方案。
   - 难题迎刃而解，全班一起庆祝一下吧！

学习编程 1

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “学习编程 1”Playground
- 投影仪或显示器

学习者所需材料：

- “收集、切换、重复”工作表
- 铅笔
- 其他纸张 (可选)

下载“学习编程”工作表

循环

概述

第 1 课：重复花瓣

- 探索：就代码中的重复步骤与现实生活的关联展开讨论
- 发现：“重复花瓣”活动
- 练习：使用循环并循环每一侧

第 2 课：障碍路线

- 探索：就循环的停止点展开讨论
- 发现：“障碍路线”活动
- 练习：拼图游戏

第 3 课：鼓乐样式

- 探索：就音乐中的重复乐段展开讨论
- 发现：“鼓乐样式”活动
- 练习：行至边缘再返回与舞步循环

学习者将习得以下技能

- 识别代码中的循环
- 将庞大的问题或任务分解成若干小步骤
- 生成一个命令序列，并使用循环重复该序列
- 测试和调试相关指令和代码

词汇

- **Loop (循环)**：重复一定次数的代码块

标准

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

探索

目标：将“循环”这一概念与现实生活联系起来。

讨论：探讨学习者在现实生活中可能会重复某项活动或步骤的次数。

示例：

- 步行
- 骑行
- 缝纫、针织或钩编

要点：循环会以你指定的次数重复某个或某组命令。

发现

目标：学习者通过制作独一无二的花瓣，开始探索“循环”的概念。

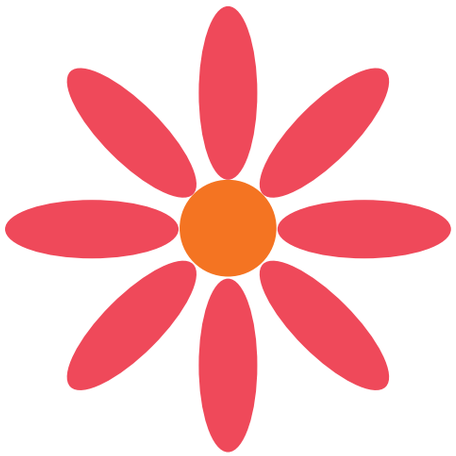
学习者所需材料：

- “重复花瓣”工作表
- 彩纸
- 铅笔
- 剪刀
- 胶棒
- 骰子

做法：

1. 学习者先在一张彩纸上画一个花瓣，花瓣长度与手掌相当，然后裁剪出来。之后将以此作为花瓣模板来制作花朵。
2. 然后每位学习者掷两次骰子，将两次掷出的数字相加，然后在“重复花瓣”工作表中将所得之和填入循环中的空白数字之处。这就是花朵将拥有的花瓣数量。
3. 学习者可利用自己的花瓣模板在彩纸上描摹花瓣，然后为花朵裁剪出相应数量的花瓣。
4. 借助“重复花瓣”工作表，学习者可拼出自己的花朵，并将各部分粘贴到相应位置上。

下载“重复花瓣”工作表



练习

目标：学习者能够在循环中编写代码以收集所有宝石。

做法：

1. 将“学习编程 1”Playground 中“for 循环”一章的介绍页面投影到屏幕上。
2. 介绍：
   - 全班一同朗读页面内容，必要时可暂停并提问。
3. 使用循环：
   - 向学习者演示传送门的工作原理，并复习 `moveForward()`、`turnLeft()`、`turnRight()` 和 `collectGem()` 命令。
   - 让学习者尝试通过不同方法引导 Byte 从开始箭头处移动到宝石处并加以收集，期间需注意哪些命令重复出现。学习者需要将命令记录在工作表或另一张纸上。
   - 收集全班学习者的想法，然后在 Swift Playgrounds app 上编写代码，引导 Byte 收集第一颗宝石并移动到传送门处。
   - 让学习者答出宝石颗数，然后将该数量添加到循环中。点按或轻点“运行我的代码”。
   - 尝试一些其他的解决方案。
   - 恭喜，Byte 闯关成功！
4. 循环每一侧：
   - 让学习者尝试通过不同方法引导 Byte 收集所有宝石，期间需注意哪些命令重复出现。
   - 要添加 `for` 循环，应采纳编辑器底部的代码建议或轻点屏幕顶部的“+”图标。
   - 收集全班学习者的想法，然后在 Swift Playgrounds 上编写代码，完成闯关。点按或轻点“运行我的代码”。
   - 试一试其他思路。
   - 恭喜，Byte 闯关成功！

学习编程 1

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “学习编程 1”Playground
- 投影仪或显示器

学习者所需材料：

- “使用循环并循环每一侧”工作表
- 铅笔
- 其他纸张 (可选)

下载“学习编程”工作表

探索

目标：探讨为何循环必须有一个明确的终点。

讨论：请学习者想象一下摩天轮或者他们熟悉的另一种游乐设施。如果游乐设施运行五轮之后仍然没有按下停止按钮，结果会怎样？让学习者想一想不停止循环的后果并要求他们再举几个例子。

要点：帮助学习者认识到，如果不停止循环，循环就会无限地重复下去。

发现

目标：学习者能够循环遍历自己设计的障碍路线，从而了解循环的工作原理。

材料：

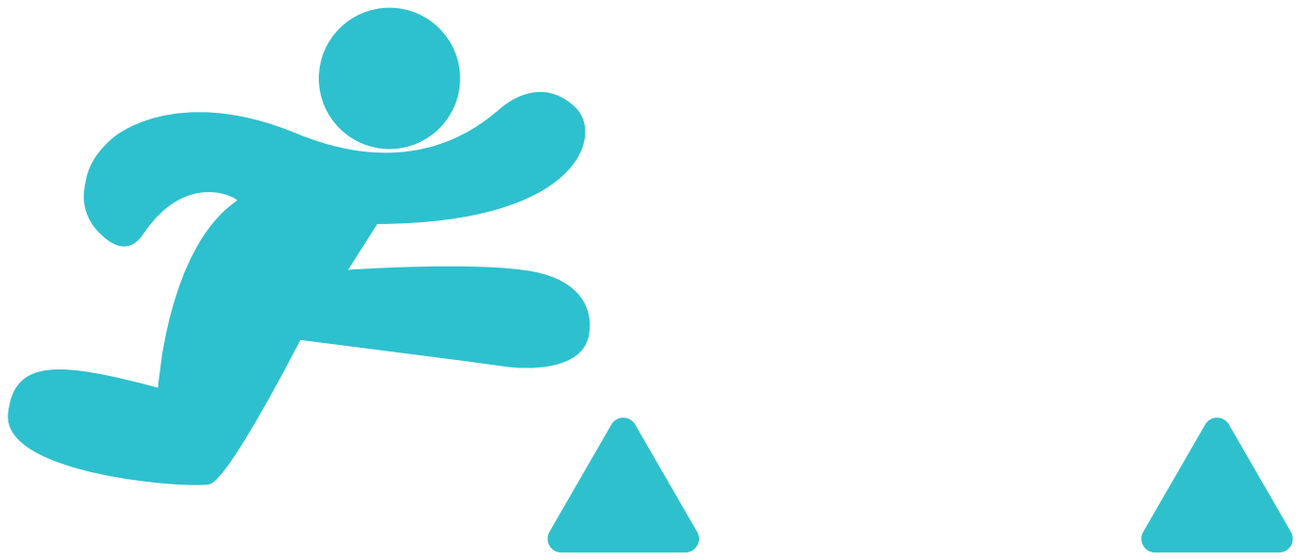
- 进行体育活动所需的空间
- 障碍路线道具
- 骰子

做法：

1. 在教室内/教室外搭建一小段障碍路线。
2. 掷一次骰子，掷出几，就让学习者穿过几次障碍路线。

替代方案：

学习者构思一连串的舞蹈动作，例如摸脚尖、跳跃、踢腿等。掷一次骰子，掷出几，就重复几次连串动作。



练习

目标：学习者能够构思出一个具有重复模式的关卡，然后以小组为单位打通关卡。

准备：学习者三人一组完成任务。使用 Painter 胶带在地面上为每组制作一个 4×4 的网格。

做法：

1. 分发材料，然后将学习者分为三人一组。
2. 阅读每个角色的介绍，然后为组内每位成员分配角色，为第一个游戏做准备。
3. 让学习者玩游戏，先从设计师角色开始。
4. 玩三次，每次轮换一遍角色卡。

角色：

- 设计师：在同伴的帮助下，将三颗宝石以某种重复模式放在网格中。将开始箭头放在网格中。
- 程序员：在同伴的帮助下，将命令卡放在网格中或网格旁，引导 Byte 移动到宝石处并加以收集。使用 Loop 卡向测试员表明命令的循环次数。
- 测试员：开始时让 Byte 位于箭头上，然后按照命令卡让 Byte 在网格中移动。如果能够收集到所有宝石，说明编程成功，恭喜你！如果没有收集到所有宝石，请一同修复代码。

替代方案：

如果你与学习者一对一教学，或者学习者在家学习，则可使用可下载的 Keynote 讲演替代活动自行玩这个游戏。

协导员所需材料：

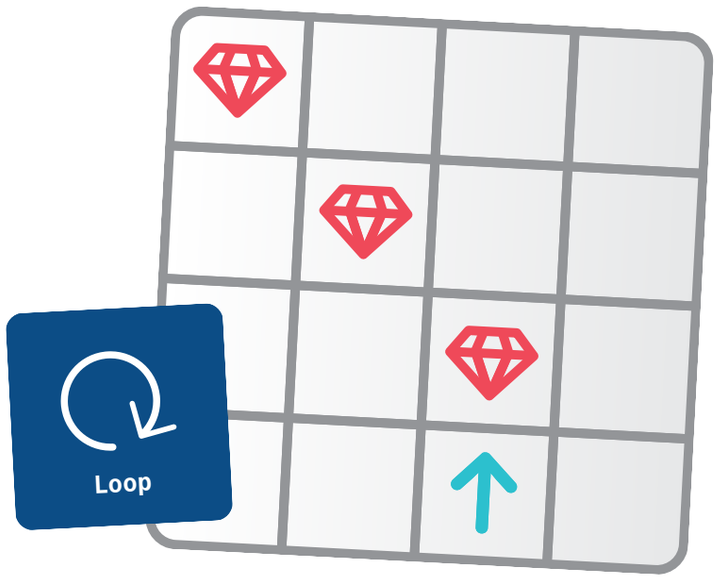
- Painter 胶带

学习者所需材料：

- 角色卡
- 命令卡：`moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `Loop`
- 宝石
- Byte
- 箭头

下载材料

下载替代活动



探索

目标：探索音乐中的重复模式。

讨论：让学习者介绍自己演奏的乐器或演唱的歌曲。询问他们在演奏或演唱时是否遇到过重复的节拍或副歌。询问他们是否可以想到歌曲或音乐中其他重复的部分。

要点：巩固“循环是由两部分组成的”这样一个概念。这两个部分分别是：

- 命令
- 重复次数

发现

目标：学习者能够重复一个鼓乐样式，将循环代码和现实生活中的某个实物关联起来。

材料：

- 一个可以敲击的对象，例如地板、大腿或书本
- 围成一圈坐下所需的空间

做法：

1. 让学习者围成一圈坐下。
2. 让学习者重复你编排的击鼓声，你举起几个手指，便重复几次。例如，如果你举起四指，学习者应当重复击鼓四次，然后停下来。
3. 学习者绕圈轮流演奏，或分小组演奏，让每人都有机会担任领鼓。

拓展：

让学习者制作鼓。

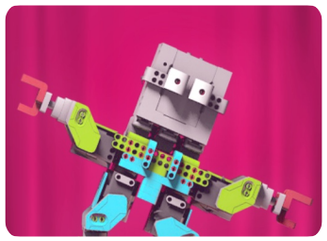
练习

目标：学习者将在一个循环内调用多个不同命令，然后确定调用循环的次数。

做法：

1. 将“学习编程 1”Playground 投影到屏幕上。导航至“For 循环”一章的第三页 (也就是“行至边缘再返回”)。
2. 行至边缘再返回：
   - 复习 `moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `toggleSwitch()` 命令。
   - 让学习者尝试通过不同方法引导 Byte 从开始箭头处移动到每个关闭的开关处并切换开关。
   - 要添加 `for` 循环，应采纳编辑器底部的代码建议或轻点屏幕顶部的“+”图标。
   - 收集全班学习者的想法，然后在 Swift Playgrounds app 上编写代码，完成闯关。点按或轻点“运行我的代码”。
   - 试一试其他思路。
   - 恭喜，Byte 闯关成功！
3. 关闭“学习编程 1”，然后打开“玩 MeeBot 学跳舞”Playground，导航到“舞步循环”页面。(“玩 MeeBot 学跳舞”Playground 页面没有学习者工作表。)
4. 舞步循环：
   - 让学习者以小组为单位，两人一组或独自使用自己的 iPad 完成循环，然后观看机器人如何跳舞。
   - 让学习者分享其舞蹈，或全班一同编排一些其他的舞蹈。
   - 和机器人一起跳起来吧！

学习编程 1



玩 MeeBot 学跳舞

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “学习编程 1”Playground
- “玩 MeeBot 学跳舞”Playground
- 投影仪或显示器

学习者所需材料：

- “行至边缘再返回”工作表
- 铅笔
- iPad 设备 (可选)
- 其他纸张 (可选)

下载“学习编程”工作表

变量

概述

第 1 课：沉没或漂浮

- 探索：就变量的更新展开讨论
- 发现：“沉没或漂浮”活动
- 练习：跟踪记录和示例游戏

第 2 课：文字游戏

- 探索：就问题的答案类型展开讨论
- 发现：“文字游戏”活动
- 练习：拼图游戏

第 3 课：自我介绍

- 探索：就根据列表回答问题展开讨论
- 发现：“自我介绍”活动
- 练习：使用循环

学习者将习得以下技能

- 将变量名称与给定值相关联
- 更改赋给变量的值
- 了解可以分配给变量的各种 Swift 类型，包括：True/False (布尔值)、数字 (整数)、文字 (字符串)、颜色 (颜色文字) 和图像 (图像文字)。
- 测试和调试相关指令和代码

词汇

- **Variable (变量)**：一个用于存储值并且可变的具名容器
- **Data (数据)**：信息
- **Boolean (布尔)**：一个取值为 `True` 或 `False` 的类型

标准

1A-AP-09、1B-AP-09、1B-AP-10、1B-AP-16 >

探索

目标：清点物品并更新可变数量，以此探讨变量的概念。

协导员所需材料：

- 白板
- 马克笔
- 橡皮擦
- 容器
- 五支铅笔 (或任何五个相同物品)

做法：

1. 首先在白板上写下一个变量语句，用来追踪你的物品。
   - 示例：`var numberOfPencils = 0`
2. 拿起一个空容器，告诉学习者这个容器表示变量 `numberOfPencils`。
3. 在容器里放入一支铅笔，然后询问学习者现在的可变数量是多少。如果他们回答正确，请擦除 0，然后写下 1。
4. 继续上述操作，直到所有铅笔均已放入容器，此时你的代码为：`var numberOfPencils = 5`。
5. 然后开始从容器中取出铅笔，一边取出一边更新变量。

要点：帮助学习者认识到变量会存储一些信息。在此示例中，变量存储的信息是一个数字，这个数字表示容器中的铅笔数量。

发现

目标：学习者能够利用找到的物品进行实验，从而确定物品是沉入水中还是漂在水面上，最后利用图像 (图像文字) 和 True/False 值 (布尔值) 记录相关数据。

学习者所需材料：

- iPad 设备
- Keynote 讲演 app
- “沉没或漂浮”工作表
- 一桶水
- 多个待测试对象

做法：

1. 将学习者分成几个小组。
2. 让他们寻找各种物品进行测试。
3. 对于每个物品，要求学习者：
   - 拍摄一张照片，然后将照片添加到工作表中。
   - 在水中进行测试。
   - 在工作表上记录测试结果：在 `True` 或 `False` 上画圈。

下载“沉没或漂浮”工作表

练习

目标：学习者能够在两种不同的编程环境下创建并更新变量。

做法：

1. 将“学习编程 2”Playground 投影到屏幕上。导航到“变量”一章。
2. 介绍：
   - 全班一同朗读页面内容，必要时可暂停并提问。
3. 跟踪记录：
   - 让学习者尝试通过不同方法引导 Hopper 从开始箭头处移动到宝石处并加以收集。学习者需要将命令记录在工作表或另一张纸上。
   - 收集全班学习者的想法，然后在 Swift Playgrounds app 上编写代码，完成闯关。点按或轻点“运行我的代码”。
   - 试一试其他思路。
   - 恭喜，Hopper 闯关成功！
4. 关闭“学习编程 2”，然后打开“石头剪刀布”Playground 的最后一页，也就是“示例游戏”。(该 Playground 页面没有学习者工作表。)
5. 示例游戏：
   - 点按或轻点“运行我的代码”，玩过游戏之后再更改内容。
   - 小组成员一同决定想要对游戏的哪些部分进行自定义。可以更改的一些有意思的内容包括 `game.roundsToWin`、`game.challenger.emoji`、`game.addOpponent` 和 `game.roundPrize`。
   - 多玩几次游戏，每次更改不同内容。

拓展：许多变量都是在 Game.swift 文件中创建的。如果学习者询问为何有些变量不是以 `var` 开头，请打开 Game.swift 文件，向他们展示游戏属性的创建位置。

学习编程 2

石头剪刀布

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “学习编程 2”Playground
- “石头剪刀布”Playground
- 投影仪或显示器

学习者所需材料：

- “跟踪记录”工作表
- 铅笔
- 其他纸张 (可选)

下载“学习编程”工作表

探索

目标：了解现实生活中回答问题时可能出现的几种答案类型，然后将这些答案类型与各 Swift 类型相关联。Swift 类型包括：是/否或对/错 (布尔值)、数字 (整数)、文字 (字符串)、颜色 (颜色文字) 和图像 (图像文字)。

协导员所需材料：

- 白板
- 马克笔

讨论：全班一起构思一些问题 (要求这些问题的答案类型不同)，然后写在白板上。

示例：

- 你的眼睛是什么颜色的？—> 颜色
- 你养宠物吗？—> 是/否
- 你有兄弟姐妹吗？—> 是/否
- 你多大？—> 数字
- 你叫什么名字？—> 文字

要点：说明变量也分为不同类型，包括数字、文字、颜色、图片以及“是/否”。根据变量的创建方式，变量类型必须保持不变，即使变量中的内容有所更新。例如，`var myAge = 8` 可以改成 `9`，但不能改成 `"nine"`。

发现

目标：学习者将通过正确填写答案类型，成功闯过文字游戏关卡。

学习者所需材料：

- “文字游戏”工作表
- 铅笔
- 彩色铅笔

做法：

将学习者分成若干小组，完成一个或多个文字游戏。每个小组最好选出至少一名读者或辅助人员。如果所有学习者都没有阅读能力，那么可以让全组一起做一些游戏。

拓展：如果学习者有阅读能力，请他们构思一个文字游戏来让同伴解开。鼓励学习者以数字、文字、颜色、图片和“是/否”等作为可填补空格的谜底。

下载“文字游戏”工作表

练习

目标：学习者能够引导 Byte 收集多颗宝石，将每颗宝石放入容器并更新一个变量。

准备：学习者三人一组完成任务。使用 Painter 胶带在地面上为每组制作一个 4×4 的网格。

做法：

1. 分发材料，然后将学习者分为三人一组。
2. 阅读每个角色的介绍，然后为组内每位成员分配角色，为第一个游戏做准备。
3. 让学习者玩游戏，先从设计师角色开始。
4. 玩三次，每次轮换一遍角色卡。

角色：

- 设计师：将多颗宝石和开始箭头放入网格。
- 程序员：在同伴的帮助下，将命令卡放在网格中或网格旁，引导 Byte 移动到宝石处并加以收集。
- 测试员：开始时让 Byte 位于箭头上，然后根据命令让 Byte 在网格中移动，将收集到的宝石放入容器。如果能够收集到所有宝石，请更新容器的变量 `numberOfGems`，然后就大功告成了！如果没有收集到所有宝石，请一同修复代码。

替代方案：

如果你与学习者一对一教学，或者学习者在家学习，则可使用可下载的 Keynote 讲演替代活动自行玩这个游戏。

协导员所需材料：

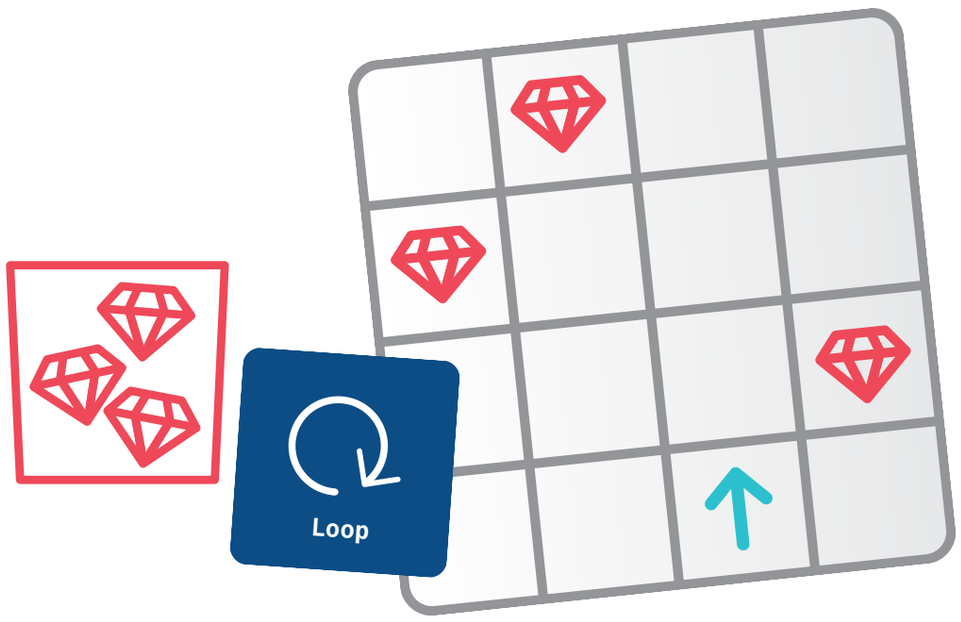
- Painter 胶带

学习者所需材料：

- 角色卡
- 命令卡：`moveForward()`、`turnLeft()`、`turnRight()`、`collectGem()` 和 `Loop`
- 宝石
- Byte
- 箭头
- 贴标容器：`var numberOfGems = _____`
- 钢笔

下载材料

下载替代活动



探索

目标：探索如何在创建变量时使用列表 (即数组)。

讨论：如果工作表要求学习者填写兄弟姐妹的姓名，但人数不止一位，情况会怎样？请全班学习者集思广益。如果有人指出可以创建一个列表，则可以告诉他们，编程人员正是这样做的！如果变量的答案不止一个，那么学习者应该创建一个列表。

让学习者提出一些可能有多个答案的问题。

示例：

- 朋友的名字 —> Rose、Sam、Joy
- 学习者的年龄 —> 7、8、7、8、7、8、9、7、8、9、8
- 最喜欢的颜色 —> 
- 最喜欢的动物 —> 

要点：学习者在代码中创建的列表就相当于是句子里列出的清单。

发现

目标：学习者能够填入变量来描述关于自己和某位同伴的信息。学习者或许可以使用数组来作为变量类型。

学习者所需材料：

- “自我介绍”和“你的简介”工作表
- 铅笔
- 彩色铅笔

做法：

1. 让学习者完成“自我介绍”工作表。
   - 如果学习者有不止一个兄弟姐妹或者养了不止一只宠物，请他们列出所有兄弟姐妹或宠物的名字并以逗号分隔。
2. 让学习者两两搭档，完成“你的简介”工作表。

替代方案：学习者可以使用其 iPad 和 Keynote 讲演来完成“你的简介”工作表，为图片答案拍照并利用格式选项为颜色文字上色。

下载“简介”工作表

练习

目标：学习者能够识别代码中的变量并想出多种方法来结合使用数组和循环。

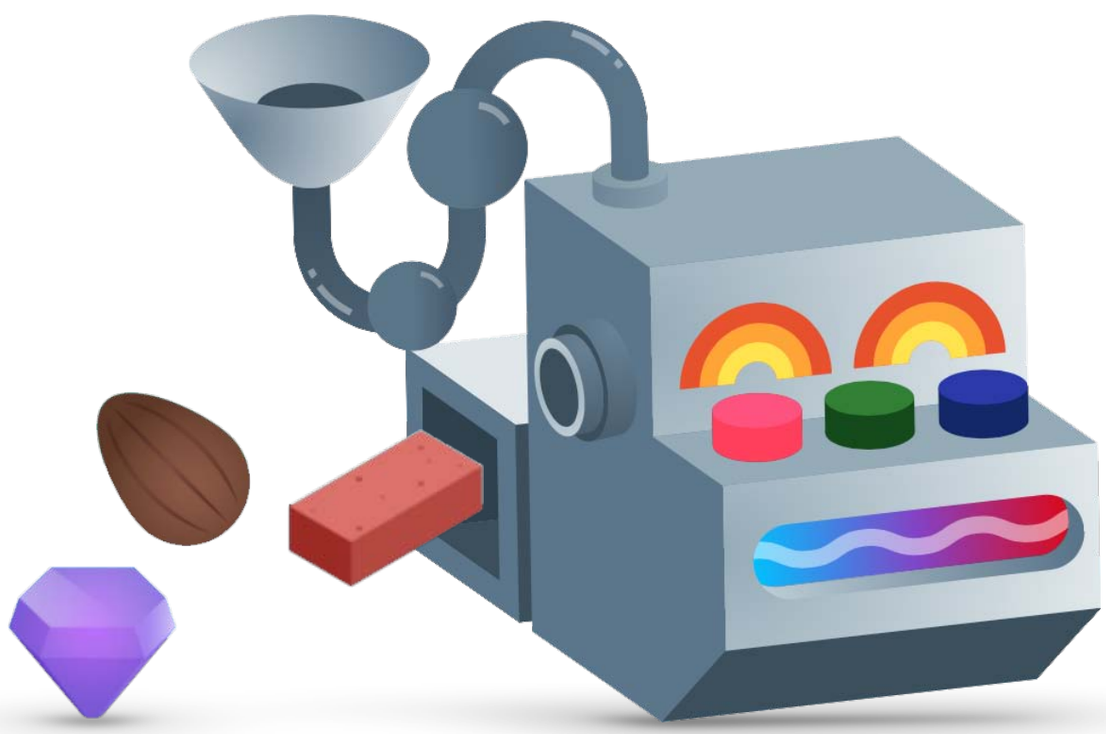
做法：

1. 将“编程机器”Playground 投影到屏幕上。
2. 介绍：
   - 全班一同朗读页面内容，必要时可暂停并提问。
   - 可选：播放前两个页面，也就是“探索编程机器”和“色彩搭配”。
3. 使用循环：
   - 学习者将在该页面上将自己对循环和变量的了解结合起来。
   - 看一看学习者能否在包含数组的代码中找出变量。
   - 点按或轻点“运行我的代码”，看一看编程机器生成的结果。
   - 继续指令中的第二个步骤，更新代码以在其中加入另一个变量、物品和嵌套循环。再次点按或轻点“运行我的代码”，看一看编程机器生成的结果。
   - 注意：请先尝试完成本页面的任务，然后再与学习者一同进行活动。

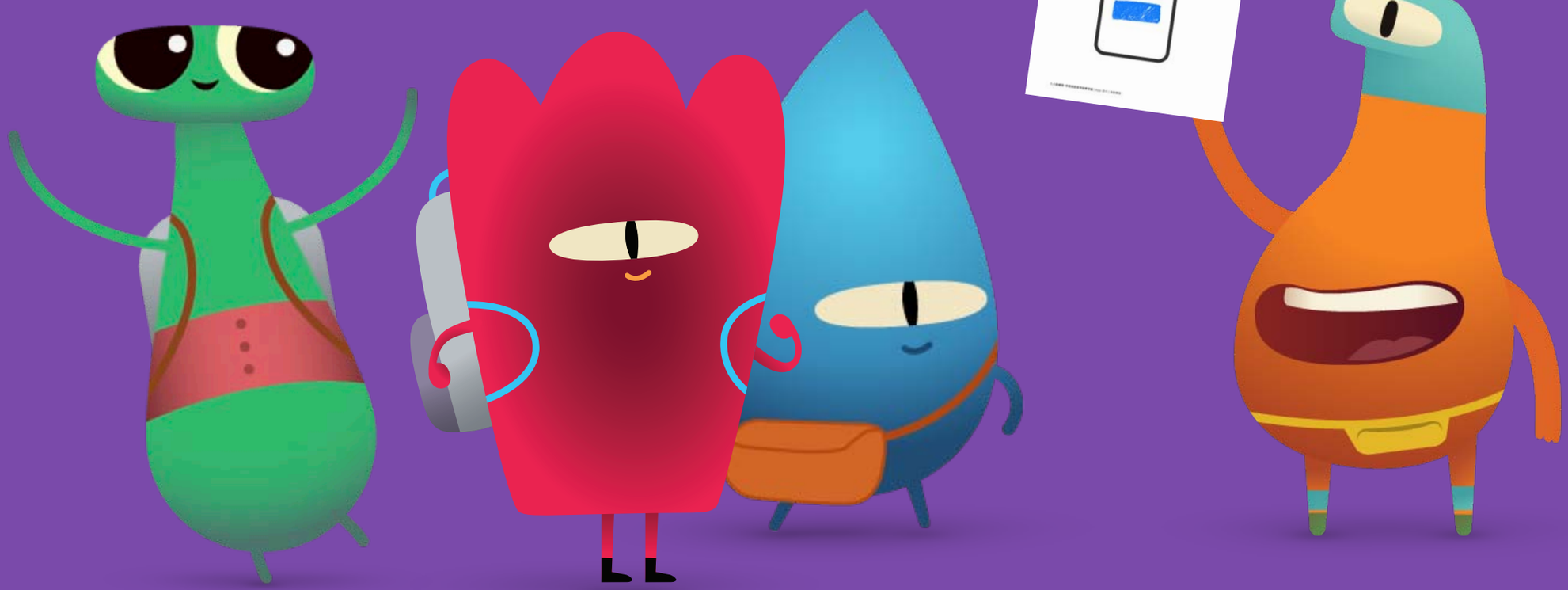
编程机器

协导员所需材料：

- iPad 或 Mac
- Swift Playgrounds app
- “编程机器”Playground
- 投影仪或显示器



App 设计



探索

目标：在各种设备上探索熟悉的 app。

做法：就学习者在家中或在学校时在 iPad 上使用的 app 展开讨论。然后学习者再介绍他们自己、父母或监护人居家时在各设备上使用的 app。

要点：巩固“app 不仅可以用在手机上，还可以用在手表、平板电脑、计算机甚至电视上”这样的概念。

拓展：深入了解一些 app 示例，询问学习者这些 app 的适用人群、主要作用和设计目的。

示例：

- App：Swift Playgrounds
- 适用人群：希望了解 Swift 的人
- 主要作用：利用关卡和课程帮助人们了解如何编程
- 设计目的：指导几乎没有或完全没有编程知识的人学会编程

发现

目标：让学习者分析一款熟悉的 app，为设计自己的 app 做好准备。

学习者所需材料：

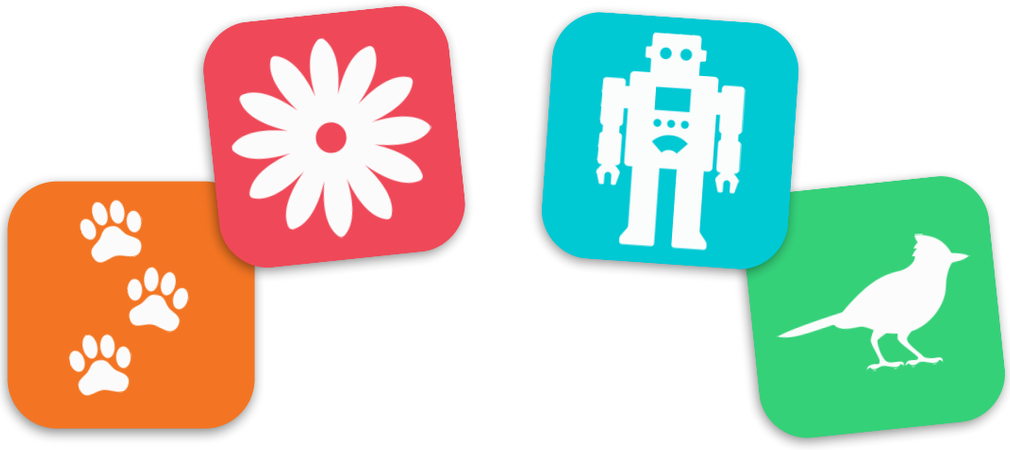
- iPad 设备
- “App 是什么？”工作表
- 铅笔
- 彩色画笔或彩色铅笔

做法：

1. 让学习者分组行动或单独行动。
2. 请学习者选择一款 iPad app。
3. 请学习者在“App 是什么？”工作表的引导下探索 app。
4. 请学习者与整个小组或同伴分享自己针对 app 得出的结论。

协导员提示：学习者年龄越小，就需要越多的帮助来完成“App 是什么？”工作表。对于 K-1 小组，请考虑让整组学习者一起探索两到三个 app。

下载“App 是什么？”工作表



练习

目标：学习者设计出自己的 app！

学习者所需材料：

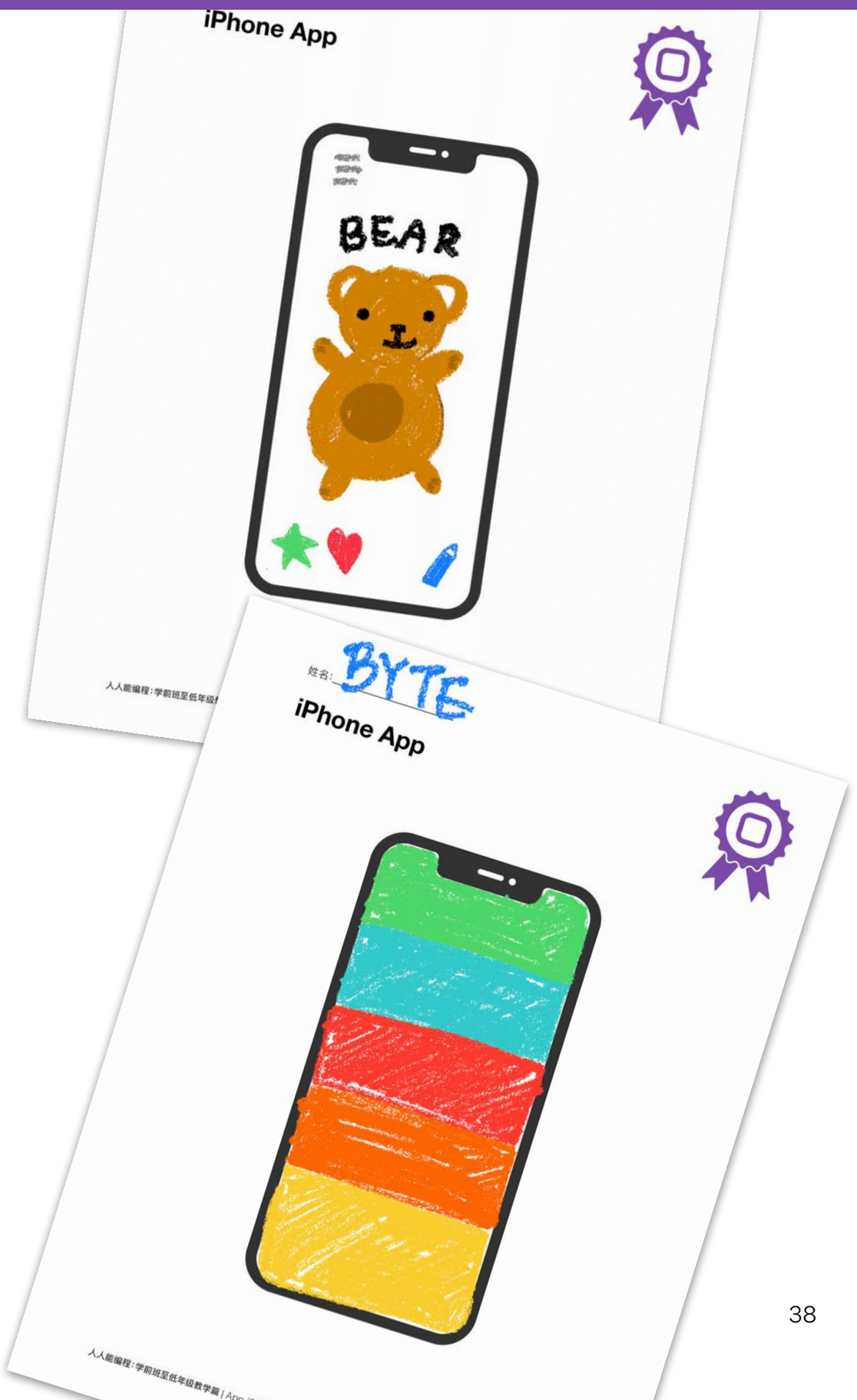
- “我的 App 设计”工作表
- 设备模板
- 其他纸张
- 铅笔
- 彩色画笔或彩色铅笔

做法：

1. 让学习者分组行动或单独行动。
2. 带领学习者了解“我的 App 设计”工作表，引导学习者迈入设计 app 的初级阶段。
3. 让学习者使用其他纸张或设备模板，制作其 app 页面的原型。
4. 指导学习者使用设备模板创建其 app 原型的最终版本。
5. 请学习者或小组逐一向大家介绍自己的想法。

下载“我的 App 设计”工作表

下载设备模板



协导员可用资源

术语表

- **Boolean (布尔)**：一个取值为 `True` 或 `False` 的类型
- **Bug (错误)**：代码中的错误
- **Command (命令)**：能够让应用程序执行特定操作的代码
- **Data (数据)**：信息
- **Debug (调试)**：找出代码中的错误并加以修复
- **Function (函数)**：一组可视需要随时运行的具名命令
- **Loop (循环)**：重复一定次数的代码块
- **Modify (修改)**：更改
- **Sequence (序列)**：事情发生的顺序
- **Step (步骤)**：一个大规模流程中的一项操作
- **Toggle (切换)**：打开或关闭
- **Variable (变量)**：一个用于存储值并且可变的具名容器

CSTA 标准 >

1A-AP

- 1A-AP-08：创建算法 (分步指令集合) 并按照算法进行操作以完成任务，为日常流程建模。
- 1A-AP-09：以数字或其他符号表示信息，为程序存储和操作数据的方式建模。
- 1A-AP-10：开发包含序列和简单循环的程序来表达想法或解决问题。
- 1A-AP-11：将解决问题需要遵循的步骤分解 (细分) 成一系列准确的指令。
- 1A-AP-12：制定方案来介绍某个程序的事件发生顺序、目标和预期结果。
- 1A-AP-14：对某个包含序列和简单循环的算法或程序进行调试 (找出并修复错误)。

1A-CS

- 1A-CS-01：选择并运行合适的软件来执行多种任务，认识到用户对于自己所用的技术存在不同的需求和偏好。

1B-AP

- 1B-AP-09：创建一些使用变量来存储和修改数据的程序。
- 1B-AP-10：创建一些包含序列、事件、循环和条件的程序。
- 1B-AP-16：扮演不同的角色，在协导员的指导下，与同伴合作完成程序开发过程中的设计、实现和审核阶段。

示例答案

接下来的几页针对每个 Swift Playground 关卡逐一提供可行的解决方案，但这些关卡的通关方法不止一种。鼓励学习者尝试通过不同方法引导 Byte 或其他人物闯关。

无论学习者选择了什么样的编程类型和目标，都应加以赞扬。有些学习者除了收集宝石之外，可能还想探索整片关卡空间，而有些学习者则可能想要一次又一次地收集宝石。别忘了，编程应该是充满乐趣的！

学习编程 1

“命令”一章 发出命令	“命令”一章 添加新命令	“函数”一章 组合新行为	“函数”一章 创建新函数

`moveForward()`
`moveForward()`
`moveForward()`
`collectGem()`
	`moveForward()`
	`moveForward()`
	`turnLeft()`
	`moveForward()`
	`moveForward()`
	`collectGem()`
		`moveForward()`
		`moveForward()`
		`moveForward()`
		`turnLeft()`
		`turnLeft()`
		`moveForward()`
		`moveForward()`
		`moveForward()`
		`collectGem()`
			`func turnRight() {`
			`  turnLeft()`
			`  turnLeft()`
			`  turnLeft()`
			`}`
			
			`moveForward()`
			`turnLeft()`
			`moveForward()`
			`turnRight()`
			`moveForward()`
			`turnRight()`
			`moveForward()`
			`turnRight()`
			`moveForward()`
			`turnLeft()`
			`moveForward()`
			`toggleSwitch()`

学习编程 1

学习编程 2

“函数”一章 收集、切换、重复	“循环”一章 使用循环	“循环”一章 循环每一侧	“循环”一章 行至边缘再返回	“变量”一章 跟踪记录

`func collectToggle() {`
`  moveForward()`
`  collectGem()`
`  moveForward()`
`  toggleSwitch()`
`  moveForward()`
`}`
 	
`collectToggle()`
`turnLeft()`
`collectToggle()`
`moveForward()`
`turnLeft()`
`collectToggle()`
`turnLeft()`
`collectToggle()`
	`for i in 1 ... 5 {`
	`  moveForward()`
	`  moveForward()`
	`  collectGem()`
	`  moveForward()`
	`}`
		`for i in 1 ... 4 {`
		`  moveForward()`
		`  collectGem()`
		`  moveForward()`
		`  moveForward()`
		`  moveForward()`
		`  turnRight()`
		`}`
			`for i in 1 ... 4 {`
			`  moveForward()`
			`  moveForward()`
			`  toggleSwitch()`
			`  turnLeft()`
			`  turnLeft()`
			`  moveForward()`
			`  moveForward()`
			`  turnLeft()`
			`}`
				`var gemCounter = 0`
				`moveForward()`
				`moveForward()`
				`collectGem()`
				`gemCounter += 1`

石头剪刀布

示例游戏

这个页面没有提供示例解决方案，因为这个游戏是完全自定义的，想怎么玩，就怎么玩！

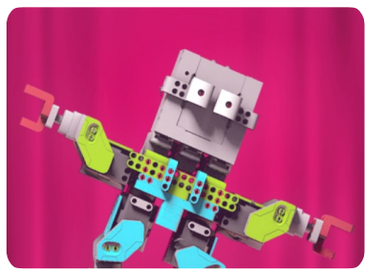
编程机器

使用循环

```
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 学跳舞

基础舞步

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

舞步循环

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

深入探讨

《人人能编程：解谜闯关教师指南》和《人人能编程：探险闯关教师指南》

教师指南旨在通过真实互动、沟通交流、团队合作、培养批判性思维和个性化学习等活动，帮助教师更从容地教授编程课程并加深学生的学习体验。教师指南还介绍了一些评估理念和技巧，帮助教师更好地区分各类班级活动。下载《解谜闯关》和《探险闯关》图书 >

人人能编程：Swift Coding Club

Swift Coding Club 为师生打开了编程教学的大门，适用于课外活动、夏令营或其他非正式学习环境。Swift Coding Club 采用模块化设计，是编程初学者的理想之选，也非常适合有编程经验的人员。下载 Swift Coding Club 套件 >

《编程快速入门》

此指南为 10 岁及以上的儿童提供了 10 个有趣的编程活动。无论在学校还是家中，学生都可使用 iPad 和 Mac 上免费的 Swift Playgrounds app 学习编程。下载《编程快速入门》>

《App 设计日志》

学生可以使用《App 设计日志》，通过 app 设计过程来解决学校或社区中存在的问题。这个日志会引导学生集思广益、进行规划、制作原型和评估他们自己的 app 创意，最后再通过宣传方案演示文稿来展示其 app 原型。下载《App 设计日志》>

Apple Teacher

Apple Teacher 计划是一项免费的、自定进度的专业学习计划，提供任意取用的学习资料和内容，启发你将 Apple 科技的力量运用于教育之中。进一步了解 >

Apple 师资培训

Apple 师资培训 Specialist 专家将指导教师参与沉浸式的实际操作活动，帮助教师积累创新教学经验并深化学生的学习体验。请致函 apla22@apple.com 进一步了解更多信息。