

ใครๆ ก็เขียนโค้ดได้

ผู้เรียนระดับเริ่มต้น

คู่มือผู้สอน



สารบัญ

บทนำ

คำสั่ง

- บทเรียนที่ 1: กิจวัตรประจำวัน
- บทเรียนที่ 2: ลำดับเรื่องราว
- บทเรียนที่ 3: ท่าเต้น

ฟังก์ชัน

- บทเรียนที่ 1: เพชรกระดาษ
- บทเรียนที่ 2: เทศกาลดนตรี
- บทเรียนที่ 3: My Calming Function

Loop

- บทเรียนที่ 1: การทำกลีบดอกชบา
- บทเรียนที่ 2: เส้นทางที่มีสิ่งกีดขวาง
- บทเรียนที่ 3: รูปแบบการตีกลอง

ตัวแปร

- บทเรียนที่ 1: ลอยหรือจม
- บทเรียนที่ 2: เกมคำศัพท์
- บทเรียนที่ 3: ทุกเรื่องเกี่ยวกับตัวฉัน

การออกแบบแอป

แหล่งข้อมูลวิทยาการ

ใครๆ ก็เขียนโค้ดได้ ผู้เรียนระดับเริ่มต้น
จัดทำขึ้นมาเพื่อช่วยนักการศึกษาและ
ครอบครัวในการแนะนำการเขียนโค้ดใน
ชั้นเรียนเด็กเล็ก ซึ่งผู้เรียนกำลังเริ่ม
พัฒนาการคิดเชิงคำนวณในขั้นต้น โดย
ผู้เรียนในระดับอนุบาลไปจนถึงประถม
ศึกษาปีที่ 3 จะได้สำรวจ ค้นพบ และเล่น
เพื่อสร้างรากฐานแนวคิดในการเขียน
โค้ดหลายๆ ผ่านบทเรียนเหล่านี้



การออกแบบการเรียนการสอน

คู่มือฉบับนี้แบ่งออกเป็น 4 โมดูล รวมไปถึงการออกแบบแอปที่เป็นโปรเจกต์ปิดท้าย ทุกโมดูลจะมี 3 บทเรียน โดยแต่ละบทเรียนจะเน้นถึงแนวคิดหนึ่งอย่างที่เกี่ยวข้องกับการเขียนโค้ด ในแต่ละบทเรียน คุณจะพบกิจกรรม 3 อย่าง ได้แก่ สำรวจ ค้นพบ และเล่น คุณสามารถแบ่งกิจกรรมต่างๆ ออกเป็นหลายเซสชันหรือหลายวันก็ได้


วันที่ 1: การอภิปรายและการเรียนรู้แบบลองใช้งานเอง

สำรวจ

แนะนำและอภิปรายถึงแนวคิดในการเขียนโค้ด

ค้นพบ


สร้างความคุ้นเคยกับแนวคิดผ่านกิจกรรมเชิงสร้างสรรค์



~25
นาที

เล่น

- เขียนโค้ดไปพร้อมๆ กับ Byte ในแอป Swift Playgrounds
- ฝึกเขียนโค้ดในกิจกรรม Keynote และเวิร์กช็อปที่มาด้วยกัน
- นำโลกของ Byte มาสู่โลกจริงด้วยเกมเขียนโค้ดไขปริศนาต่างๆ แบบไม่ต้องใช้อุปกรณ์



~25
นาที

วันที่ 2: การเชื่อมโยงการเรียนรู้เข้ากับโค้ด

ขอบเขตและลำดับ

เราออกแบบ 4 โมดูลในคู่มือฉบับนี้เพื่อนำไปใช้กับชั้นเรียนระดับอนุบาลจนถึงประถมศึกษาปีที่ 3 และสามารถนำไปใช้ในลำดับใดก็ได้ เราขอแนะนำให้คุณใช้โมดูล "การออกแบบแอป" ในเวลาใดก็ได้ หรือมากกว่า 1 ครั้งต่อปี เมื่อผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับโค้ดและแอปเพิ่มมากขึ้น

ตัวอย่าง:

ระดับชั้น	โมดูล	โครงการปิดท้าย	เวลารวมโดยประมาณ
อนุบาล	คำสั่ง	การออกแบบแอป	4 ชั่วโมง
ประถม 1	ฟังก์ชัน	การออกแบบแอป	4 ชั่วโมง
ประถม 2	Loop	การออกแบบแอป	4 ชั่วโมง
ประถม 3	ตัวแปร	การออกแบบแอป	4 ชั่วโมง

เรียนรู้ต่อ

สำหรับการสอนชั้นประถมศึกษาปีที่ 4 ไปจนถึงมัธยมศึกษาปีที่ 2 หลักสูตร "ใครๆ ก็เขียนโค้ดไฮปรีตนาได้" พร้อมกับคู่มือการแสดงผลงานแอปและบันทึกการออกแบบแอปมีเนื้อหาให้เรียนรู้มากกว่า 45 ชั่วโมง ดูเพิ่มเติมในคู่มือหลักสูตร "ใครๆ ก็เขียนโค้ดได้"



พอร์ตโฟลิโอของผู้เรียน (ไม่บังคับ)

ตลอดโมดูลเหล่านี้ ให้รวบรวมวัตถุจากกิจกรรมต่างๆ เพื่อสร้างพอร์ตโฟลิโอร่วมกับผู้เรียนของคุณ











โมดูล	บทเรียนที่	สิ่งของแนะนำ
คำสั่ง	กิจวัตรประจำวัน	<ul style="list-style-type: none"> เวิร์กชีต "การออกคำสั่ง" เวิร์กชีต "การเพิ่มคำสั่งใหม่"
	ลำดับเรื่องราว	<ul style="list-style-type: none"> ภาพประเด็นเรื่องราวของลำดับเรื่องราว ภาพกลุ่มลำดับเรื่องราว
	ท่าเต้น	<ul style="list-style-type: none"> การ์ดท่าเต้น วิดีโอท่าเต้น (ไม่บังคับ)
ฟังก์ชัน	เพชรกระดาษ	<ul style="list-style-type: none"> รูปร่างของเพชรกระดาษ เวิร์กชีตการเขียนลักษณะการทำงานใหม่ เวิร์กชีตการสร้างฟังก์ชันใหม่
	เทศกาลดนตรี	<ul style="list-style-type: none"> ฟังก์ชันที่เขียนหรือวิดีโอคอนเสิร์ตเทศกาลดนตรี
	My Calming Function	<ul style="list-style-type: none"> ภาพวาดหรือวิดีโอ My Calming Function เวิร์กชีต "เก็บ สลับเปิดปิด ทำซ้ำ"
Loops	การทำกลีบดอกช้ำ	<ul style="list-style-type: none"> การทำกลีบดอกช้ำ เวิร์กชีตการใช้ Loop เวิร์กชีตการใช้ Loop ทุกด้าน
	เส้นทางที่มีสิ่งกีดขวาง	<ul style="list-style-type: none"> วิดีโอหรือรูปภาพเส้นทางที่มีสิ่งกีดขวาง (ไม่บังคับ)
	รูปแบบการตีกลอง	<ul style="list-style-type: none"> เวิร์กชีต "ไปจนสุดขอบแล้วกลับมา" วิดีโอหรือรูปภาพการตีกลอง (ไม่บังคับ)
ตัวแปร	ลอยหรือจม	<ul style="list-style-type: none"> ลอยหรือจม เวิร์กชีต "คอยติดตามดู"
	เกมคำศัพท์	<ul style="list-style-type: none"> เกมคำศัพท์
	ทุกเรื่องเกี่ยวกับตัวฉัน	<ul style="list-style-type: none"> ทุกเรื่องเกี่ยวกับตัวฉัน ทุกเรื่องเกี่ยวกับตัวคุณ
การออกแบบแอป		<ul style="list-style-type: none"> แอปคืออะไร การออกแบบแอปของฉัน ต้นแบบการออกแบบแอป

เริ่มต้นใช้งาน Swift Playgrounds ใน iPad หรือ Mac



ก่อนที่จะศึกษาบทเรียนนี้ ให้แน่ใจว่าคุณได้ดาวน์โหลด [Swift Playgrounds](#), [Pages](#) และ [Keynote](#) แล้ว โมดูลในคู่มือฉบับนี้ใช้ชุดสนามเด็กเล่นที่แตกต่างกันไป สิ่งที่คุณต้องมีในแต่ละโมดูลมีดังนี้

โมดูล	Playgrounds	วิธีการดาวน์โหลดใน Swift Playgrounds
คำสั่ง	 เรียนรู้วิธีเขียนโค้ด 1  MeeBot Dances	เพื่อเป็นการสมัครสมาชิกฟีด MeeBot Playgrounds ให้เลื่อนไปที่ด้านล่างของหน้าจอ "สนามเด็กเล่นเพิ่มเติม" แล้วแตะ "ป้อน URL สมัครสมาชิก" จากนั้นป้อน: ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json
ฟังก์ชัน	 เรียนรู้วิธีเขียนโค้ด 1	
Loop	 เรียนรู้วิธีเขียนโค้ด 1  MeeBot Dances	เพื่อเป็นการสมัครสมาชิกฟีด MeeBot Playgrounds ให้เลื่อนไปที่ด้านล่างของหน้าจอ "สนามเด็กเล่นเพิ่มเติม" แล้วแตะ "ป้อน URL สมัครสมาชิก" จากนั้นป้อน: ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json
ตัวแปร	 เรียนรู้วิธีเขียนโค้ด 2  เป่ายิ่งดูบ  เครื่องเขียนโค้ด	"เป่ายิ่งดูบ" และ "เครื่องเขียนโค้ด" สามารถดูเครื่องเขียนโค้ดได้ในส่วนหนังสือของหน้าจอ "สนามเด็กเล่นเพิ่มเติม"

การออกแบบแอป

ดูความต้องการขั้นต่ำสำหรับ Swift Playgrounds ใน [App Store](#) ไปที่ [บริการช่วยเหลือของ Apple](#) เพื่อขอความช่วยเหลือเกี่ยวกับ Swift Playgrounds

เคล็ดลับสำหรับวิทยากร

เพื่อให้ได้ประโยชน์สูงสุดจากบทเรียนร่วมกับผู้เรียน ให้ลองเคล็ดลับบางส่วนเหล่านี้

กิจกรรมการสำรวจและค้นพบ:

- แก้ไขไวยากรณ์หรือตัวอักษรพิเศษให้เรียงง่ายขึ้นเมื่อเขียนหรือแสดงโค้ด เช่น
 - 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 = ■

กิจกรรมการเล่น:

- เพื่อให้แอป Swift Playgrounds เรียบง่ายขึ้นสำหรับผู้เรียนระดับเริ่มต้น ให้ทำตามคำแนะนำในแผนของบทเรียนต่างๆ รวมถึง
 - อ่านคำแนะนำร่วมกันในกลุ่มใหญ่
 - ให้คำแนะนำที่เรียบง่ายกับผู้เรียนสำหรับเวิร์กช็อปที่มาด้วยกันเพื่อให้พวกเขาคิดโซลูชันของตัวเองได้
 - ใช้ iPad หรือ Mac ของวิทยากร 1 เครื่องในการไขปริศนาในแอป
- let และ var: ในคู่มือฉบับนี้ไม่ได้มีเนื้อหาเกี่ยวกับคีย์เวิร์ด let เพื่อไม่ให้เกิดความสับสนใน Swift Playgrounds โปรดเปลี่ยนคีย์เวิร์ด let ทั้งหมดเป็นคีย์เวิร์ด var ก่อนแสดงหน้าต่างต่างๆ ให้กับผู้เรียน ในสนามเด็กเล่นที่เราแนะนำ คีย์เวิร์ดทั้ง 2 สามารถใช้แทนกันได้
 - let = ตัวแปรไม่มีการเปลี่ยนแปลง
 - var = ตัวแปรมีการเปลี่ยนแปลง

ส่วนเพิ่มเติม:

- ต่อยอดกิจกรรมการเล่นเกม ด้วยการเพิ่มทักษะด้านตัวเลข ทักษะการอ่านเขียน คำที่พบเห็นบ่อย การสะกดคำ และอีกมากมาย ลองกิจกรรมการเล่นเกมที่ง่าย ๆ ในหลักสูตรฟังค์ชันเพื่อสร้างแรงบันดาลใจ
- ปรับแต่งกิจกรรมการเล่นเกมที่ง่าย ๆ ให้ตรงความต้องการด้วยการให้ผู้เรียน สร้างการ์ดแทนคำสั่งต่างๆ เช่น twirl() หรือ jump()



หน้าบทนำ



หน้าสนามเด็กเล่น

← คำสั่ง



ภาพรวม

บทเรียนที่ 1: กิจกรรมประจำวัน

- **สำรวจ:** การอภิปรายที่เชื่อมโยงการอบขนมเข้ากับคำสั่งต่างๆ
- **ค้นพบ:** กิจกรรมกิจวัตรประจำวัน
- **เล่น:** การส่งคำสั่งและการเพิ่มคำสั่งใหม่

บทเรียนที่ 2: ลำดับเรื่องราว

- **สำรวจ:** การอภิปรายที่เชื่อมโยงลำดับเรื่องราวเข้ากับคำสั่งต่างๆ
- **ค้นพบ:** กิจกรรมลำดับเรื่องราว
- **เล่น:** เกมปริศนาต่างๆ

บทเรียนที่ 3: ทำเต็น

- **สำรวจ:** การอภิปรายที่เชื่อมโยงทำเต็นเข้ากับคำสั่งต่างๆ
- **ค้นพบ:** กิจกรรมทำเต็น
- **เล่น:** สวัสดิ์ MeeBot และทำเต็นพื้นฐาน

ผู้เรียนจะสามารถ

- ใช้ตัวอย่างจากชีวิตประจำวันในการอธิบายถึงคำสั่งแบบทีละขั้นตอน
- ใส่คำสั่งตามลำดับเพื่อให้ดูเหมาะสม
- ทดสอบและดีบักคำสั่งและโค้ด

คำศัพท์

- **ลำดับ:** ลำดับของสิ่งที่เกิดขึ้น
- **ขั้นตอน:** การดำเนินการหนึ่งอย่างในกระบวนการที่ใหญ่กว่าปกติ
- **ปรับเปลี่ยน:** การเปลี่ยนแปลง
- **คำสั่ง:** โค้ดที่สั่งให้แอปพลิเคชันดำเนินการอย่างใดอย่างหนึ่ง
- **บัก:** ข้อผิดพลาดในโค้ด
- **ดีบัก:** การค้นหาและแก้ไขข้อผิดพลาดในโค้ด

มาตรฐาน

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

สำรวจ

วัตถุประสงค์: แนะนำแนวคิดเรื่องคำสั่ง โดยเชื่อมโยงเข้ากับการอบขนมบราวนี่

การอภิปราย:

- พวกเขาทำขนมบราวนี่ตามสูตรหรือไม่
- พวกเขาทำขั้นตอนต่างๆ ของสูตรตามลำดับหรือไม่

สิ่งที่ควรจำ: แต่ละขั้นตอนหรือวิธีการในสูตรเป็นเหมือนกับคำสั่งในโค้ด ให้ผู้เรียนนึกถึงคำสั่งของตัวเอง

ค้นพบ

วัตถุประสงค์: จำลองกระบวนการของกิจวัตรประจำวันด้วยการระบุคำสั่งแบบทีละขั้นตอน

สื่อ: การ์ด Washing Your Hands

คำแนะนำ:

1. สับการ์ด Washing Your Hands และหงายให้ดูบนโต๊ะหรือติดไว้บนกระดาน การ์ดไม่ควรเรียงลำดับกัน
2. ถามผู้เรียนว่าพวกเขาคิดว่ามีบีก้อยู่ในลำดับการล้างมือของตนเองหรือไม่
3. ให้ผู้เรียนทำการดีบัก หรือแก้ไขคำสั่ง ด้วยการย้ายการ์ดทีละใบไปยังตำแหน่งที่ถูกต้อง

ทางเลือก:

ให้ผู้เรียนจับคู่หรือกลุ่มเล็กๆ และให้ชุดการ์ดกับแต่ละกลุ่ม

ส่วนเพิ่มเติม:

ให้ผู้เรียนนึกถึงชุดคำสั่งแบบทีละขั้นตอนของตนเองสำหรับกิจกรรมบางอย่างที่พวกเขาทำทุกวัน และทำภาพที่แสดงถึงขั้นตอนแต่ละอย่าง

↓ [ดาวน์โหลดการ์ด Washing Your Hands](#)



เล่น

วัตถุประสงค์: ผู้เรียนจะสามารถเพิ่มคำสั่งในลำดับที่ถูกต้องเพื่อเก็บเพชรเม็ดแรกใน "เรียนรู้วิธีเขียนโค้ด 1" ในแอป Swift Playgrounds

คำแนะนำ:

1. ฉายหน้าบทนำของบท "คำสั่ง" ในสนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1" ขึ้นบนหน้าจอ
2. บทนำ:
 - อ่านเนื้อหาในหน้าต่างๆ ร่วมกันในชั้น และหยุดถามคำถาม หากต้องการ
3. การออกคำสั่ง:
 - ทบทวนคำสั่ง 2 อย่างที่ผู้เรียนจะต้องมีเพื่อพา Byte ไปเก็บเพชร นั่นคือ `moveForward()` และ `collectGem()`
 - ให้ผู้เรียนทดลองวิธีการต่างๆ ในการนำทาง Byte จากลูกศรเริ่มต้นไปยังเพชร และทำการเก็บ ลองบันทึกคำสั่งต่างๆ ไว้ในเวิร์กชิตหรือในแผ่นกระดาษแยกต่างหาก
 - รวบรวมไอเดียจากชั้นเรียน และเขียนโค้ดลงในแอป Swift Playgrounds เพื่อไขปริศนาคลิกหรือแตะ "รันโค้ดของฉัน"
 - ลองไอเดียที่แตกต่างกันหลายๆ อย่าง
 - จลองไปกับ Byte

ส่วนเพิ่มเติม:

หากผู้เรียนพร้อม ให้ไปยังหน้าถัดไป "การเพิ่มคำสั่งใหม่" ผู้เรียนจะได้ใช้คำสั่งใหม่ `turnLeft()` ที่นี้



เรียนรู้วิธีเขียนโค้ด 1

สื่อสำหรับวิทยากร:

- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1"
- โปรเจ็กเตอร์หรือจอภาพ

สื่อสำหรับผู้เรียน:

- เวิร์กชิตการส่งคำสั่งและการเพิ่มคำสั่งใหม่
- ดินสอ
- กระดาษเพิ่มเติม (ไม่บังคับ)

↓ [ดาวน์โหลดเวิร์กชิต "เรียนรู้วิธีเขียนโค้ด"](#)



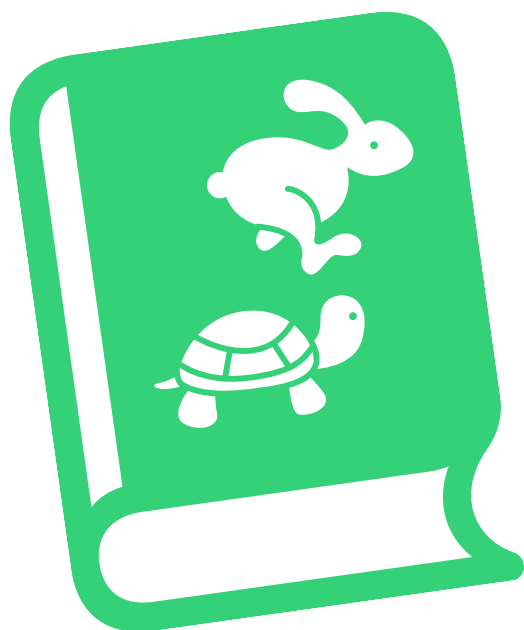
สำรวจ

วัตถุประสงค์: สำรวจว่าการลำดับเรื่องราว (ตอนต้นถึงกลางเรื่องและตอนจบ) ของหนังสือเป็นอย่างไรเพื่อให้เรื่องราวดูสมเหตุสมผล

การอภิปราย:

- ถามผู้เรียนว่าหนังสือเรียงตามลำดับหรือไม่
- จะเป็นอย่างไรหากตอนต้น ตอนกลาง และตอนจบของหนังสือไม่เป็นไปตามลำดับ
- สำรวจตัวอย่างต่างๆ

สิ่งที่ควรจำ: สร้างการเชื่อมโยงกับโค้ด โดยเน้นถึงความสำคัญในการกำหนดคำสั่งในการเขียนโค้ดตามลำดับที่ถูกต้อง เช่นเดียวกับเรื่องราวๆ ของเนื้อเรื่อง

**ค้นพบ**

วัตถุประสงค์: หลังจากสร้างภาพของประเด็นต่างๆ จากเรื่องราวได้แล้ว ผู้เรียนจะสามารถใส่ภาพตามลำดับเพื่อถ่ายทอดเรื่องราวอีกครั้งได้อย่างแม่นยำ

สื่อสำหรับวิทยากร:

- ไวท์บอร์ด
- ปากกามาร์คเกอร์

สื่อสำหรับผู้เรียน:

- กระดาษ
- ปากกามาร์คเกอร์หรือดินสอสี
- ทางเลือก: iPad และแอปวาดรูป

คำแนะนำ:

1. อ่านเรื่องราวที่ผู้เรียนรู้จักดี พิจารณาประเด็นเรื่องราวหลักในเรื่องราวร่วมกันในชั้นเรียน ควรให้นึกถึงเรื่องราว 4 ถึง 6 ประเด็น
2. สร้างกลุ่มเล็กๆ ที่มีจำนวนผู้เรียนเท่ากับจำนวนเรื่องราว เช่น หากมีเรื่องราว 4 ประเด็น ก็จะมีผู้เรียน 4 คนต่อกลุ่ม
3. ให้ผู้เรียนแต่ละคนในกลุ่มวาดเรื่องราวมา 1 เรื่อง
4. กลุ่มต่างๆ ผัดกันมายืนหน้าห้อง โดยให้ผู้เรียนถือภาพเรื่องราวของตนไว้โดยไม่เรียงตามลำดับ
5. ผู้ชมช่วยจัดลำดับภาพใหม่ โดยขยับทีละภาพ
6. ถ่ายรูปของแต่ละกลุ่มไว้เมื่อผู้เรียนอยู่ในลำดับที่ถูกต้อง

ส่วนเพิ่มเติมหรือทางเลือก:

ให้ผู้เรียนแต่ละกลุ่มทำในเรื่องราวอื่น โดยพิจารณาถึงเรื่องราวร่วมกันในกลุ่มก่อนการวาดภาพ

เล่น

วัตถุประสงค์: ผู้เรียนจะสามารถนำทาง Byte ผ่านตารางจริงไปยังเพชรโดยใช้คำสั่งด้านทิศทางได้

การเตรียมตัว: ผู้เรียนจะร่วมงานกันเป็นกลุ่มละ 3 คน ใช้เทปกาวเพื่อสร้างตารางแบบ 4x4 ไว้บนพื้นให้กับแต่ละกลุ่ม

คำแนะนำ:

1. แจกสื่อการเรียนรู้ และแบ่งผู้เรียนออกเป็น 3 กลุ่ม
2. อ่านแต่ละบทบาทอย่างละเอียด และกำหนดบทบาทให้กับแต่ละคนในกลุ่มสำหรับเกมแรก
3. ให้ผู้เรียนเล่นเกม โดยเริ่มต้นด้วยบทบาทนักร้อง
4. เล่น 3 ครั้ง โดยที่มีการหมุนเวียนการ์ดบทบาทในแต่ละครั้ง

บทบาท:

- นักร้อง: วางเพชรและลูกศรเริ่มต้นไว้บนตาราง
- โปรแกรมเมอร์: ด้วยความช่วยเหลือจากเพื่อนของคุณ วางการ์ดคำสั่งไว้บนหรือถัดจากตารางดังกล่าว เพื่อนำทาง Byte ไปยังเพชรและทำการเก็บ
- ผู้ทดสอบ: เริ่มต้นโดยให้ Byte อยู่ที่ลูกศร ทำตามการ์ดคำสั่งเพื่อให้ Byte เดินไปตามตาราง หากคุณเก็บเพชรได้ ก็ลองกันได้เลย แต่หากไม่ได้ ให้ร่วมมือกันในทีมเพื่อดีบักหรือแก้ไขโค้ด

ทางเลือก:

หากผู้เรียนกำลังทำงานกับคุณคนเดียวหรือเรียนที่บ้าน พวกเขาสามารถเล่นเกมด้วยตัวเองได้โดยใช้กิจกรรม Keynote ทางเลือกที่มีให้ดาวน์โหลด

สื่อสำหรับวิทยากร:

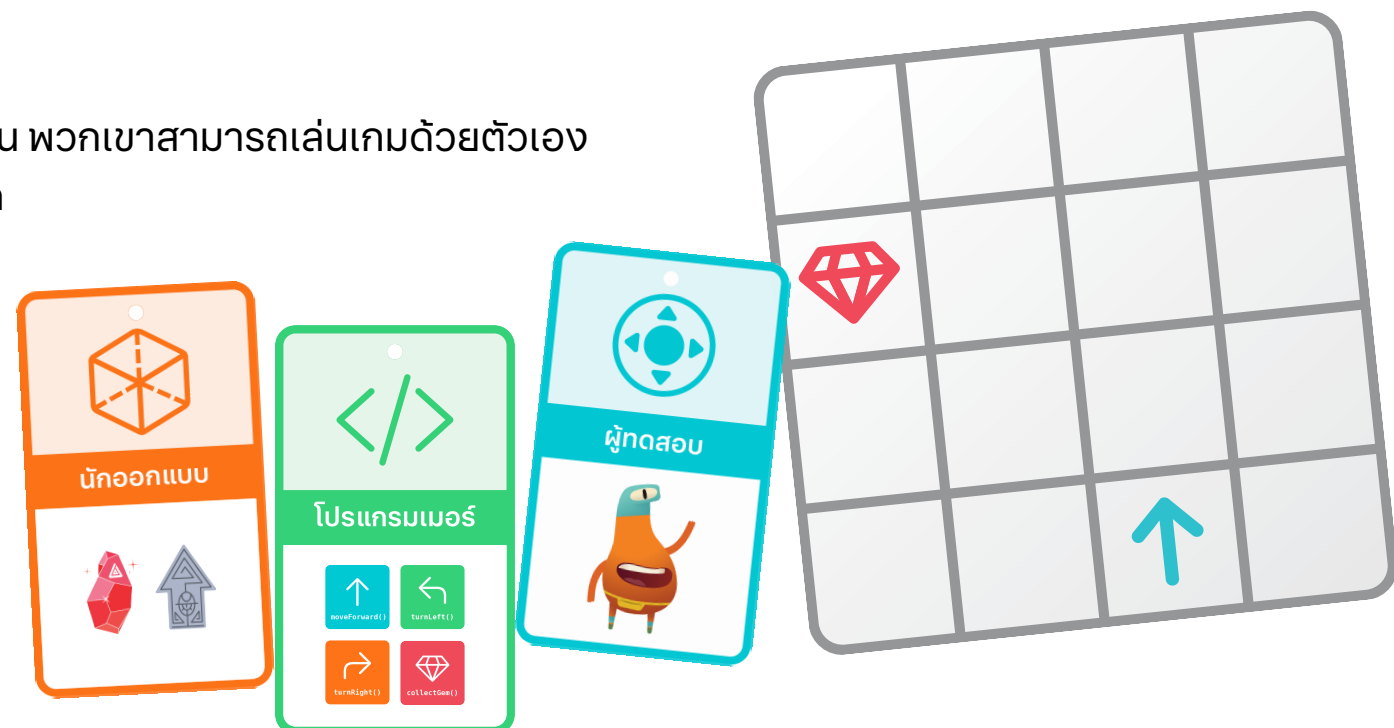
- เทปกาว

สื่อสำหรับผู้เรียน:

- การ์ดบทบาท
- การ์ดคำสั่ง: `moveForward()`, `turnLeft()`, `turnRight()` และ `collectGem()`
- เพชร
- Byte
- ลูกศร

↓ ดาวน์โหลดสื่อ

↓ ดาวน์โหลดกิจกรรมทางเลือก



สำรวจ

วัตถุประสงค์: สำรวจโอเดย์ที่ว่าเราใส่ความคิดสร้างสรรค์ลงในการเขียนโค้ดได้

การอภิปราย:

- ถามผู้เรียนว่าเคยเรียนการเต้นมาก่อนหรือไม่
- การเต้นมีลำดับขั้นตอนที่ต้องทำตามหรือไม่
- พวกเขารู้ได้อย่างไรว่าต้องทำอะไรต่อ
- ทำเต็นมีชื่อเป็นของตัวเองหรือไม่
- ผู้เรียนเคยใช้ท่าเดิมๆ ในการเต้น หรือในรูปแบบการเต้นที่แตกต่างกันหรือไม่

สิ่งที่ควรจำ: ช่วยให้ผู้เรียนเชื่อมโยงว่าเราใส่ความคิดสร้างสรรค์ลงในการเขียนโค้ดได้ เช่นเดียวกันกับการออกแบบท่าเต้น คนเขียนโค้ดสามารถสร้างคำสั่งใหม่ๆ และใส่เข้าไว้ด้วยกันในรูปแบบที่น่าสนใจและแตกต่างกันได้

ค้นพบ

วัตถุประสงค์: คิดท่าเต้นสั้นๆ พร้อมกับการ์ดที่แสดงถึงท่าเต้นต่างๆ การ์ดท่าเต้นแต่ละใบเป็นเหมือนคำสั่งในสนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด"

สื่อสำหรับผู้เรียน:

- iPad
- แอป Keynote
- แอปกล้อง
- พื้นที่สำหรับเต้น

คำแนะนำ:

1. ให้ผู้เรียนจับคู่หรือแบ่งเป็นกลุ่มเล็กๆ และคิดท่าเต้นสั้นๆ
2. เมื่อผู้เรียนคิดท่าเต้นได้แล้ว พวกเขาจะต้องทำการ์ดท่าเต้นท่าต่างๆ ผู้เรียนต้องเพิ่มภาพและชื่อของท่าเต้นลงในการ์ดแต่ละใบ โดยทำให้สร้างสรรค์และชวนขบขันที่สุดเท่าที่จะทำได้
3. แต่ละกลุ่มแสดงท่าเต้นของตน แล้วสนุกไปกับการเต้นร่วมกันทั้งชั้นเรียน

ทางเลือก:

ผู้เรียนสามารถใช้การ์ดท่าเต้นที่มีให้ดาวน์โหลดด้านล่างในการสร้างท่าเต้นของตน หรือใช้การ์ดเป็นตัวอย่างในขณะที่สร้างการ์ดของตัวเอง

ส่วนเพิ่มเติม:

ผู้เรียนทำวิดีโอการเต้นของตัวเองเพื่อแสดงให้กลุ่มดู

↓ [ดาวน์โหลดการ์ดท่าเต้น](#)



เล่น

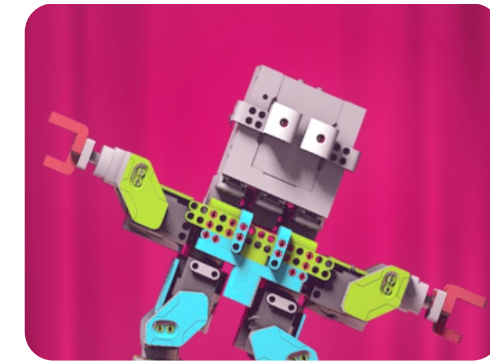
วัตถุประสงค์: สร้างลำดับขั้นตอนในการสอนท่าเต้นใหม่ให้กับหุ่นยนต์ MeeBot

คำแนะนำ:

1. ฉายสนามเด็กเล่น MeeBot Dances ขึ้นบนหน้าจอ คุณจะต้องสมัครสมาชิกสนามเด็กเล่น หากว่ายังไม่ได้สมัคร
2. บทนำ:
 - อ่านเนื้อหาในหน้าต่างๆ ร่วมกันในชั้น และหยุดถามคำถามหากต้องการ
3. สวัสดิ์ MeeBot:
 - คลิกหรือแตะ "รันโค้ดของฉัน" และดูการเต้นของหุ่นยนต์
4. ท่าเต้นพื้นฐาน:
 - ผู้เรียนจะเลือกคำสั่ง 8 อย่างจากรายการแนะนำร่วมกันเป็นกลุ่ม เป็นคู่ หรือคนเดียวบน iPad และดูการเต้นของหุ่นยนต์
 - ให้ผู้เรียนแชร์การเต้นของตัวเอง หรือสร้างการเต้นที่แตกต่างออกไปร่วมกันในชั้นเรียน
 - ออกสแต็ปการเต้นด้วยหุ่นยนต์

ส่วนเพิ่มเติม:

- ไปที่หน้าถัดไป "ท่าเต้น" ผู้เรียนจะสามารถเพิ่มท่าเต้นในฟังก์ชัน myDanceRoutine () ได้ มากน้อยตามคำสั่งที่ตนเองต้องการที่นี้



MeeBot Dances

สื่อสำหรับวิทยากร:

- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น MeeBot Dances
- โปรเจ็กเตอร์หรือจอภาพ

สื่อสำหรับผู้เรียน:

- อุปกรณ์ iPad (ไม่บังคับ)



ฟังก์ชัน



ภาพรวม

บทเรียนที่ 1: เพชรกระดาด

- สำรอง: การอธิบายเกี่ยวกับคำแนะนำแบบทีละขั้นตอน
- ค้นพบ: กิจกรรมเพชรกระดาด
- เล่น: การเขียนลักษณะการทำงานใหม่และการสร้างฟังก์ชันใหม่

บทเรียนที่ 2: เทศกาลดนตรี

- สำรอง: การอธิบายเกี่ยวกับวิธีการตั้งชื่อฟังก์ชัน
- ค้นพบ: กิจกรรมเทศกาลดนตรี
- เล่น: เกมปริศนาต่างๆ

บทเรียนที่ 3: My Calming Function

- สำรอง: การอธิบายเกี่ยวกับการแก้ปัญหาในวิธีที่หลากหลาย
- ค้นพบ: กิจกรรม My Calming Function
- เล่น: เก็บ สลับเปิดปิด ทำซ้ำ

ผู้เรียนจะสามารถ

- แยกงานหรือปัญหาใหญ่ออกเป็นขั้นตอนย่อยที่เล็กลง
- สร้างชุดขั้นตอนในการแก้ไขปัญหหรือทำงานให้เสร็จสิ้น
- ตั้งชื่อฟังก์ชัน
- ทดสอบและดีบักโค้ด

คำศัพท์

- **ฟังก์ชัน:** ชุดคำสั่งที่มีชื่อ ซึ่งสามารถนำไปใช้ได้เมื่อต้องการ
- **สลับเปิดปิด:** เปิดหรือปิด

มาตรฐาน

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

สำรวจ

วัตถุประสงค์: สำรวจไอดีของการรวมชุดคำสั่งและการตั้งชื่อ

การอภิปราย: ให้ทั้งชั้นเรียนร่วมกันตัดสินใจว่าจะเน้นกิจกรรมประจำวันใดให้ผู้เรียนระบุชื่อกิจกรรมประจำวันของตนและขั้นตอนดำเนินการ

ตัวอย่าง: กิจกรรมก่อนเข้านอน

- ขั้นตอนที่ 1: แปรงฟัน
- ขั้นตอนที่ 2: เข้าห้องน้ำ
- ขั้นตอนที่ 3: อ่านหนังสือ
- ขั้นตอนที่ 4: บอกลาค่อนนอน
- ขั้นตอนที่ 5: ปิดไฟ

สิ่งที่ควรจำ: การคิดหาชุดคำสั่งและการตั้งชื่อเป็นแนวคิดแบบเดียวกันกับการสร้างฟังค์ชั่น

ส่วนเพิ่มเติม: ถามผู้เรียนว่าคำสั่งสำหรับขั้นตอนต่างๆ ของตนนั้นสามารถทำให้มีความเฉพาะเจาะจงมากขึ้นได้หรือไม่ ตัวอย่างเช่น ขั้นตอนโดยเฉพาะเจาะจงในการแปรงฟันของคุณเป็นอย่างไร

ค้นพบ

วัตถุประสงค์: ผู้เรียนจะเริ่มต้นด้วยการทำตามคำสั่งในการพับเพชรกระดาษ แล้วเขียนหรือวาดคำสั่งสำหรับการพับรูปร่างอื่นๆ ที่ตัวเองต้องการ

สื่อสำหรับผู้เรียน:

- กระดาษ
- กรรไกร
- ดินสอ
- อุปกรณ์ iPad (ไม่บังคับ)

คำแนะนำ:

แสดงวิธีการพับเพชรกระดาษให้ผู้เรียนดู

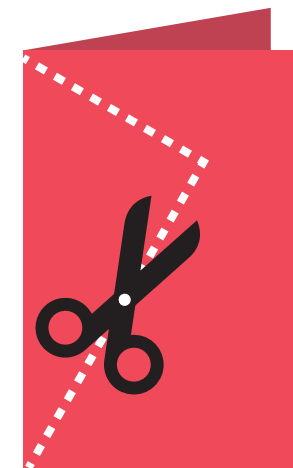
1. พับกระดาษครึ่งหนึ่ง
2. ลากเส้นจากมุมบนของด้านที่พับมาที่ระยะ 1 หรือ 2 นิ้วเหนือกึ่งกลางกระดาษ
3. ลากอีกเส้นจากจุดปลายของเส้นแรกไปที่มุมล่างของด้านที่พับ
4. ตัดตามเส้นที่คุณลากไว้
5. นำเพชรออกจากเศษกระดาษ แล้วคลี่ออก

ขอให้ผู้เรียนพับรูปร่างในแบบของตัวเอง

1. แบ่งผู้เรียนออกเป็นกลุ่มเล็กๆ
2. ให้แต่ละกลุ่มเลือกรูปร่างที่จะพับ
3. ให้เวลากับผู้เรียนในการฝึกพับรูปร่างดังกล่าวหนึ่งหรือสองรอบ
4. ให้ผู้เรียนเขียนหรือวาดคำสั่งสำหรับการพับรูปร่างดังกล่าว แล้วตั้งชื่อคำสั่งนั้นๆ เช่น "Make a Circle" หรือ "The Letter T"

ทางเลือก:

ทำวิดีโอแสดงวิธีการพับรูปร่างของตนเอง



เล่น

วัตถุประสงค์: จากการทำงานเป็นกลุ่ม ผู้เรียนจะสามารถแยกย่อยขั้นตอนต่างๆ ที่จำเป็นในการทำให้ Byte รวบรวมเพชร

คำแนะนำ:

1. ฉายรูปสนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1" ขึ้นบนหน้าจอ ไปที่บท "ฟังค์ชั่น" ใน "เรียนรู้วิธีเขียนโค้ด 1"
2. บทนำ:
 - อ่านเนื้อหาในหน้าต่างๆ ร่วมกันในชั้น และหยุดถามคำถาม หากต้องการ
3. การเขียนลักษณะการทำงานใหม่:
 - ทบทวนคำสั่งต่างๆ `moveForward()`, `turnLeft()` และ `collectGem()` - โดยจำไว้เสมอว่าคุณไม่มีคำสั่ง `turnRight()`
 - ให้ผู้เรียนทดลองวิธีการต่างๆ ในการนำทาง Byte จากลูกศรเริ่มต้นไปยังเพชรและทำการเก็บลองบันทึกรหัสคำสั่งต่างๆ ไว้ในเวิร์กชีตหรือในแผ่นกระดาษแยกต่างหาก
 - รวบรวมไอเดียจากชั้นเรียน และเขียนโค้ดลงในแอป Swift Playgrounds เพื่อไขปริศนาคlickหรือแตะ "รันโค้ดของฉัน"
 - ลองไอเดียที่แตกต่างกันหลายๆ อย่าง
 - จลองไปกับ Byte
4. การสร้างฟังค์ชั่นใหม่:
 - ให้ผู้เรียนนึกไอเดียเกี่ยวกับการสร้างฟังค์ชั่น `turnRight()` โดยใช้สิ่งที่ได้เรียนรู้จาก "การเขียนลักษณะการทำงานใหม่" ในหน้าสนามเด็กเล่นหน้าสุดท้าย
 - ให้ผู้เรียนทดลองวิธีการต่างๆ ในการนำทาง Byte จากลูกศรเริ่มต้นไปยังสวิตช์ที่ปิดอยู่แล้วทำการเปิด โดยใช้ฟังค์ชั่น `turnRight()` ของตัวเอง
 - รวบรวมไอเดียจากชั้นเรียน และเขียนโค้ดลงในแอป Swift Playgrounds เพื่อไขปริศนาคlickหรือแตะ "รันโค้ดของฉัน"
 - ลองไอเดียที่แตกต่างกันหลายๆ อย่าง
 - จลองช้ไปกับ Byte - นั่นเป็นปริศนาที่ยากพอตัวเลย



เรียนรู้วิธีเขียนโค้ด 1

สื่อสำหรับวิทยากร:

- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1"
- โปรเจ็กเตอร์หรือจอภาพ

สื่อสำหรับผู้เรียน:

- เวิร์กชีต "การเขียนลักษณะการทำงานใหม่" และ "การสร้างฟังค์ชั่นใหม่"
- ดินสอ
- กระดาษเพิ่มเติม (ไม่บังคับ)

↓ [ดาวน์โหลดเวิร์กชีต "เรียนรู้วิธีเขียนโค้ด"](#)

สำรวจ

วัตถุประสงค์: ปรับใช้ความรู้ด้านคำสั่ง และฟังค์ชั่นต่างๆ กับเพลงด้วยการตั้งชื่อที่สื่อความหมายได้

การอภิปราย: ให้ผู้เรียนนึกถึงเพลงต่างๆ และตั้งชื่อแต่ละเพลงเป็นชื่อฟังค์ชั่นที่สื่อความหมายได้

ตัวอย่าง: ชื่อเรียกฟังค์ชั่นสำหรับเพลง "Twinkle, Twinkle, Little Star" อาจเป็น `singTwinkle()` แต่ `singSong1()` เป็นชื่อที่ไม่เหมาะนัก เพราะเพลงแรกอาจมีการเปลี่ยนแปลงได้

สิ่งที่ควรรำ: การตั้งชื่อฟังค์ชั่นด้วยชื่อที่สื่อความหมายได้เป็นสิ่งสำคัญ เพราะช่วยให้คุณและผู้อื่นเข้าใจโค้ดดังกล่าวได้ง่ายขึ้น

ค้นพบ

วัตถุประสงค์: ผู้เรียนจะสร้างคอนเสิร์ตด้วยการเรียกใช้คำสั่งเพลงที่แตกต่างกันในฟังค์ชั่นคอนเสิร์ต

สื่อสำหรับวิทยาการ:

- iPad หรือ Mac
- โปรเจกเตอร์หรือจอภาพ
- ไวท์บอร์ด
- ปากกามาร์คเกอร์

คำแนะนำ:

1. ช่วยผู้เรียนตั้งชื่อฟังค์ชั่นสำหรับเพลงต่างๆ เช่น `singHappyBirthday()`
2. ให้กลุ่มเลือกลำดับเพลงที่จะร้อง
3. เขียนนิยามฟังค์ชั่นสำหรับคอนเสิร์ต และใส่ฟังค์ชั่นพร้อมกับคำสั่งของเพลง

ตัวอย่าง:

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

ทางเลือก:

ผู้เรียนร้องเพลงในกลุ่มเล็ก โดยให้แต่ละกลุ่มนึกถึงรายชื่อเพลง ชื่อฟังค์ชั่นของเพลง และลำดับการร้องเพลงของตน แล้วให้แต่ละกลุ่มเล่นเพลงของตัวเองและบันทึกวิดีโอคอนเสิร์ตของตนไว้

เล่น

วัตถุประสงค์: ผู้เรียนจะแก้สมการง่ายๆ วางเพชรไว้ที่คำตอบ และพา Byte เดินผ่านตารางโดยใช้คำสั่งด้านทิศทางได้

การเตรียมตัว: ผู้เรียนจะร่วมงานกันเป็นกลุ่มละ 3 คน ใช้เทปขาวเพื่อสร้างตารางแบบ 4x4 ไว้บนพื้นให้กับแต่ละกลุ่ม วางลูกศรเริ่มต้นไว้ในสี่เหลี่ยมช่องหนึ่ง และวางตัวเลข 1 ตัวไว้ในแต่ละช่องที่เหลือ

คำแนะนำ:

1. แจกสื่อการเรียน และแบ่งผู้เรียนออกเป็น 3 กลุ่ม
2. อ่านแต่ละบทบาทอย่างละเอียด และกำหนดบทบาทให้กับแต่ละคนในกลุ่มสำหรับเกมแรก
3. ให้ผู้เรียนเล่นเกม โดยเริ่มต้นด้วยบทบาทนักออกแบบ
4. เล่น 3 ครั้ง โดยที่มีการหมุนเวียนการ์ดบทบาทในแต่ละครั้ง

บทบาท:

- นักออกแบบ: ทอยลูกเต๋า 2 ลูก ด้วยความช่วยเหลือจากเพื่อนของคุณ บวกตัวเลข 2 ตัวเข้าด้วยกัน และวางเพชรไว้บนตารางที่มีผลรวมดังกล่าว
- โปรแกรมเมอร์: ด้วยความช่วยเหลือจากเพื่อนของคุณ วางการ์ดคำสั่งไว้บนหรือถัดจากตารางดังกล่าว เพื่อนำทาง Byte ไปยังเพชรและทำการเก็บ
- ผู้ทดสอบ: เริ่มต้นโดยให้ Byte อยู่ที่ลูกศร ทำตามการ์ดคำสั่งเพื่อให้ Byte เดินไปตามตาราง หากคุณเก็บเพชรได้ ก็ลองกันได้เลย แต่หากไม่ได้ ให้ร่วมมือกันในทีมเพื่อแก้ไขโค้ด

ทางเลือก:

หากผู้เรียนกำลังทำงานกับคุณคนเดียวหรือเรียนที่บ้าน พวกเขาสามารถเล่นเกมด้วยตัวเองได้โดยใช้กิจกรรม Keynote ทางเลือกที่มีให้ดาวน์โหลด

สื่อสำหรับวิทยากร:

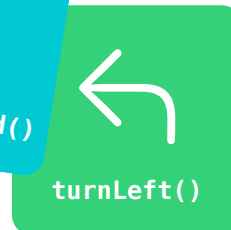
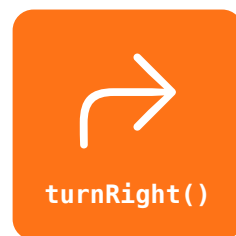
- เทปขาว
- ชุดตัวเลขสำหรับแต่ละตาราง

สื่อสำหรับผู้เรียน:

- การ์ดบทบาท
- การ์ดคำสั่ง: `moveForward()`, `turnLeft()`, `turnRight()` และ `collectGem()`
- เพชร
- Byte
- ลูกศร
- ลูกเต๋า 2 ลูก

↓ ดาวน์โหลดสื่อ

↓ ดาวน์โหลดกิจกรรมทางเลือก

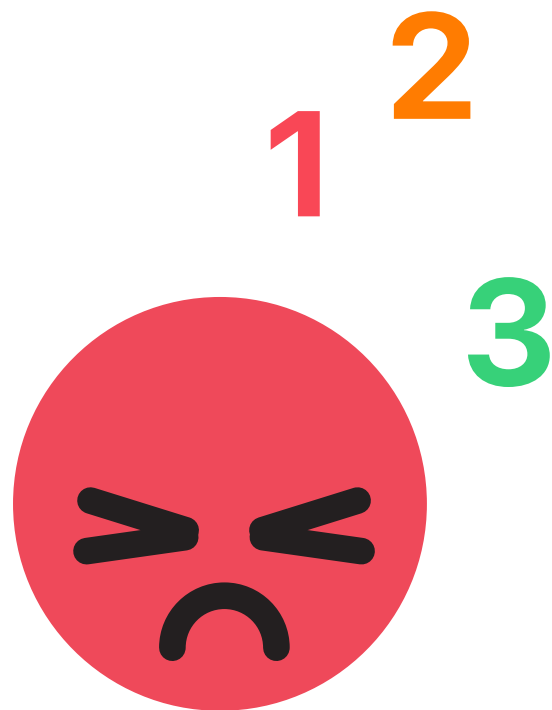


สำรวจ

วัตถุประสงค์: ผู้เรียนจะเข้าใจว่ามีวิธีแก้ปัญหามากกว่าหนึ่งวิธีเสมอ

การอภิปราย: ให้ผู้เรียนคิดถึงปัญหาที่ตนเองมี แล้วแบ่งปันวิธีที่ตนใช้แก้ปัญหาตามกลุ่มดังกล่าวว่ามีคนที่สามารถแก้ปัญหาได้ด้วยวิธีการอื่นหรือไม่ สำรวจปัญหาและวิธีแก้ปัญหาก็แตกต่างกันหลายๆ อย่าง

สิ่งที่ควรจำ: ช่วยผู้เรียนในการสร้างการเชื่อมโยงกับการเขียนโค้ด และเรียนรู้ว่ามีวิธีแก้ไขปัญหาด้านการเขียนโปรแกรมมากกว่าหนึ่งวิธีเสมอ

**ค้นพบ**

วัตถุประสงค์: ผู้เรียนจะเขียนฟังก์ชันเทคนิคการทำให้สงบและตั้งชื่อฟังก์ชันนั้น

สื่อสำหรับผู้เรียน:

- เวอร์กชิต My Calming Function
- ดินสอ
- ปากกาหรือดินสอสี

คำแนะนำ:

เคล็ดลับ: หากทำได้ การให้ผู้เรียนทำกิจกรรมนี้ด้วยตัวคนเดียวจะให้ผลดีที่สุด

1. ขอให้ผู้เรียนระดมความคิดเกี่ยวกับวิธีที่พวกเขาทำใจของตนเองให้สงบ เวลาที่พวกเขาอารมณ์ไม่ดีที่บ้านหรือที่โรงเรียน ให้พวกเขาแยกย่อยเทคนิคการทำให้สงบออกเป็นขั้นตอน
2. แจกเวอร์กชิต My Calming Function และให้ผู้เรียนวาดขั้นตอนของเทคนิคการทำให้สงบ
3. ให้ผู้เรียนตั้งชื่อเทคนิคการทำให้สงบ พวกเขาสามารถใช้ Camel Case ได้ เช่น `countToTen()` หรือแค่ประโยคสั้นๆ อย่าง "Count to ten"

ส่วนเพิ่มเติม:

ไม่ใช่อุปกรณ์: ให้ผู้เรียนแสดงเทคนิคการทำให้สงบในกลุ่มเล็กๆ หรือหน้าชั้นเรียน

ใช้ iPad: ผู้เรียนจัดทำวิดีโอเทคนิคการทำให้สงบเพื่อแบ่งปันในห้องเรียน



ดาวน์โหลดเวอร์กชิต My Calming Function

เล่น

วัตถุประสงค์: ผู้เรียนจะสามารถเขียนฟังก์ชันที่ประกอบด้วยคำสั่งประเภทต่างๆ แล้วใช้ฟังก์ชันดังกล่าวในการแก้ปริศนา

คำแนะนำ:

1. ฉายหน้า "เก็บ สลับเปิดปิด ทำซ้ำ" ในสนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1" ขึ้นบนหน้าจอ โดยเน้นถึงฟังก์ชันเปล่าที่ผู้เรียนจะต้องช่วยทำให้สมบูรณ์
2. เก็บ สลับเปิดปิด ทำซ้ำ:
 - ทบทวนคำสั่ง `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` และ `toggleSwitch()`
 - ให้ผู้เรียนลองระบุส่วนต่างๆ ของปริศนาที่ซ้ำ แล้วใช้ไอเดียของพวกเขาในการทำให้ฟังก์ชันในแอปสมบูรณ์และตั้งชื่อ
 - ให้ผู้เรียนคิดสัญลักษณ์สำหรับฟังก์ชัน และบันทึกสัญลักษณ์และชื่อฟังก์ชันดังกล่าวลงในปุ่มคำสั่งบนเวิร์กชิต
 - ด้วยคำสั่งเพิ่มเติม ผู้เรียนจะได้ทดลองวิธีการต่างๆ ในการนำทาง Byte ไปเก็บรวบรวมเพชรทุกชิ้นและสับสวิตช์ทั้งหมด ลองบันทึกคำสั่งต่างๆ ไว้ในเวิร์กชิตหรือในแผ่นกระดาษแยกต่างหาก
 - รวบรวมไอเดียจากชั้นเรียน และเขียนโค้ดลงในแอป Swift Playgrounds เพื่อไขปริศนาคลิกหรือแตะ "รันโค้ดของฉัน"
 - ลองวิธีการที่แตกต่างกันหลายๆ อย่าง
 - จลองร่วมกันในชั้นเรียน - นั่นเป็นปริศนาที่ยากพอตัวเลย



เรียนรู้วิธีเขียนโค้ด 1

สื่อสำหรับวิทยากร:

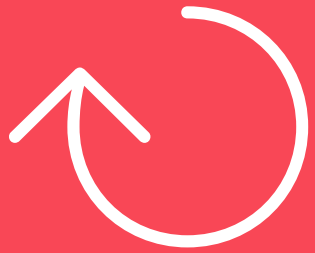
- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1"
- โปรเจกเตอร์หรือจอภาพ

สื่อสำหรับผู้เรียน:

- เวิร์กชิต "เก็บ สลับเปิดปิด ทำซ้ำ"
- ดินสอ
- กระดาษเพิ่มเติม (ไม่บังคับ)

↓ [ดาวน์โหลดเวิร์กชิต "เรียนรู้วิธีเขียนโค้ด"](#)





Loop



ภาพรวม

บทเรียนที่ 1: การทำลูปดอกซ้ำ

- สำรอง: การอภิปรายเกี่ยวกับการทำขั้นตอนต่างๆ ในโค้ดให้เกิดขึ้นจริง
- ค้นพบ: กิจกรรมการทำลูปดอกซ้ำ
- เล่น: การใช้ Loop และ Loop ทุกด้าน

บทเรียนที่ 2: เส้นทางที่มีสิ่งกีดขวาง

- สำรอง: การอภิปรายเกี่ยวกับจุดหยุดใน Loop
- ค้นพบ: กิจกรรมเส้นทางที่มีสิ่งกีดขวาง
- เล่น: เกมปริศนาต่างๆ

บทเรียนที่ 3: รูปแบบการติกลง

- สำรอง: การอภิปรายเกี่ยวกับ Loop ในทางดนตรี
- ค้นพบ: กิจกรรมรูปแบบการติกลง
- เล่น: "ไปจนสุดขอบแล้วกลับมา" และ Dance Loops

ผู้เรียนจะสามารถ

- ระบุ Loop ในโค้ด
- แยกงานหรือปัญหาใหญ่ออกเป็นขั้นตอนย่อยที่เล็กลง
- สร้างลำดับคำสั่งและทำลำดับนั้นซ้ำโดยใช้ Loop
- ทดสอบและดีบักคำสั่งและโค้ด

คำศัพท์

- **Loop:** โค้ดชุดหนึ่งที่จะรันซ้ำๆ ตามจำนวนครั้งที่กำหนด

มาตรฐาน

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

สำรวจ

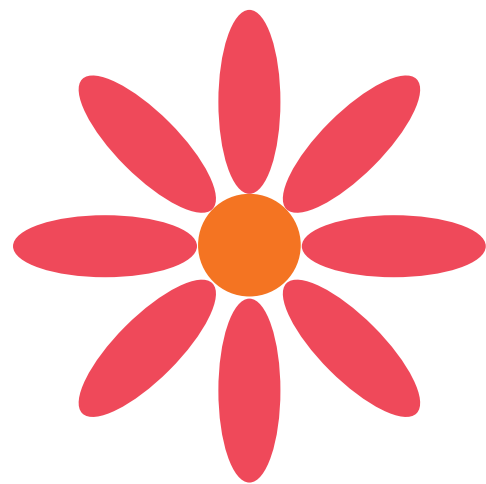
วัตถุประสงค์: เชื่อมโยงแนวคิดของ Loop กับชีวิตจริง

การอภิปราย: สำรวจจำนวนครั้งที่ผู้เรียนอาจทำงานหรือขั้นตอนซ้ำในชีวิตจริง

ตัวอย่าง:

- การเดิน
- การขี่จักรยาน
- การเย็บผ้า การถักไหมพรม หรือการถักโครเชต์

สิ่งที่ควรจำ: Loop ทำซ้ำคำสั่งหรือชุดคำสั่งตามจำนวนที่คุณกำหนด



ค้นพบ

วัตถุประสงค์: ผู้เรียนจะเริ่มต้นสำรวจหลักการของ Loop ด้วยการทำดอกไม้ที่ไม่เหมือนใคร

สื่อสำหรับผู้เรียน:

- เวอร์กชิตการทำกลีบดอกซ้ำ
- กระดาษสี
- ดินสอ
- กรรไกร
- กาวแท่ง
- ลูกเต๋า

คำแนะนำ:

1. ผู้เรียนวาดกลีบดอกมาหนึ่งกลีบ มีความยาวประมาณฝ่ามือของตนเอง ไว้บนกระดาษสีและตัดออกมา กระดาษนี้จะเป็นแม่แบบสำหรับกลีบของดอกไม้ของพวกเขา
2. ผู้เรียนแต่ละคนจะกอลูกเต๋า 2 ลูก บวกตัวเลขเข้าด้วยกัน และใส่ลงในตัวเลขที่หายไป Loop บนเวอร์กชิตการทำกลีบดอกซ้ำของตัวเอง ตัวเลขนี้เป็นจำนวนกลีบที่ดอกไม้จะมี
3. ผู้เรียนจะใช้แม่แบบสำหรับกลีบดอกในการร่างกลีบดอกของตัวเองบนกระดาษสี และตัดออกมาตามจำนวนกลีบดอกที่ถูกต้องสำหรับดอกไม้ของตัวเอง
4. ผู้เรียนจะใช้เวอร์กชิตการทำกลีบดอกซ้ำในการประกอบดอกไม้ของตัวเองขึ้นมา และติดกาวส่วนต่างๆ ให้เข้าที่



ดาวน์โหลดเวอร์กชิตการทำกลีบดอกซ้ำ



เล่น

วัตถุประสงค์: ผู้เรียนจะสามารถเขียนโค้ดในลูบเพื่อเก็บเพชรทั้งหมด

คำแนะนำ:

1. ฉายหน้าบทนำของบท "For Loop" ในสนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1" ขึ้นบนหน้าจอ
2. บทนำ:
 - อ่านเนื้อหาในหน้าต่างๆ ร่วมกันในชั้น และหยุดถามคำถาม หากต้องการ
3. การใช้ Loop:
 - แสดงวิธีการทำงานของฟอร์ทอลล์ให้กับผู้เรียน และทบทวนคำสั่ง `moveForward()`, `turnLeft()`, `turnRight()` และ `collectGem()`
 - ให้ผู้เรียนทดลองวิธีการต่างๆ ในการนำทาง Byte จากลูกศรเริ่มต้นไปยังเพชรและทำการเก็บ พร้อมกับสังเกตว่ามีคำสั่งใดที่ใช้ซ้ำ ลองบันทึกคำสั่งต่างๆ ไว้ในเวิร์กชิตหรือในแผ่นกระดาษแยกต่างหาก
 - รวบรวมไอเดียจากทั้งชั้นเรียน และเขียนโค้ดลงในแอป Swift Playgrounds เพื่อนำทาง Byte ไปเก็บเพชรเม็ดแรกและเดินเข้าฟอร์ทอลล์
 - ถามผู้เรียนว่ามีเพชรอยู่ที่เม็ด และเพิ่มตัวเลขดังกล่าวลงใน Loop คลิกหรือแตะ "รันโค้ดของฉัน"
 - ลองวิธีการที่แตกต่างกันหลายๆ อย่าง
 - จลองชัยไปกับ Byte
4. การใช้ Loop ทุกด้าน:
 - ให้ผู้เรียนทดลองวิธีการต่างๆ ในการเก็บรวบรวมเพชรทั้งหมด พร้อมกับสังเกตว่ามีคำสั่งใดที่ใช้ซ้ำ
 - ในการเพิ่ม For Loop ให้ใช้คำแนะนำโค้ดที่ด้านล่างของตัวแก้ไขหรือ + ที่ด้านบนหน้าจอ
 - รวบรวมไอเดียจากชั้นเรียน และเขียนโค้ดลงใน Swift Playgrounds เพื่อไขปริศนาคลิกหรือแตะ "รันโค้ดของฉัน"
 - ลองไอเดียที่แตกต่างกันหลายๆ อย่าง
 - จลองไปกับ Byte

**เรียนรู้วิธีเขียนโค้ด 1****สื่อสำหรับวิทยากร:**

- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1"
- โปรเจ็กเตอร์หรือจอภาพ

สื่อสำหรับผู้เรียน:

- เวิร์กชิตการใช้ Loop และ Loop ทุกด้าน
- ดินสอ
- กระดาษเพิ่มเติม (ไม่บังคับ)

↓ [ดาวน์โหลดเวิร์กชิต "เรียนรู้วิธีเขียนโค้ด"](#)

สำรวจ

วัตถุประสงค์: สำรวจว่าทำไม Loop ถึงต้องมีจุดสิ้นสุดที่เฉพาะเจาะจงเสมอ

การอภิปราย: ขอให้ผู้เรียนนึกภาพถึงชิงช้าสวรรค์หรือเครื่องเล่นอื่นที่พวกเขาคุ้นเคย จะเกิดอะไรขึ้น หากคนคุมเครื่องเล่นไม่กดปุ่มหยุดหลังจากผ่านไปแล้ว 5 รอบ ให้ผู้เรียนนึกถึงตัวอย่างอื่นๆ ของสิ่งที่จะเกิดขึ้น หากไม่มีการหยุด Loop

สิ่งที่ควรจำ: ช่วยให้ผู้เรียนเข้าใจว่า หากพวกเขาไม่หยุด Loop ก็จะมีการทำซ้ำไปไม่สิ้นสุด

ค้นพบ

วัตถุประสงค์: ผู้เรียนจะพบรูปแบบการทำงานของ Loop ด้วยการ Loop ผ่านเส้นทางที่มีสิ่งกีดขวางที่พวกเขาออกแบบ

วัตถุประสงค์:

- พื้นที่ในการทำกิจกรรม
- พร้อมสำหรับเส้นทางที่มีสิ่งกีดขวาง
- ลูกเต๋า

คำแนะนำ:

1. สร้างเส้นทางที่มีสิ่งกีดขวางสั้นๆ ในชั้นเรียนของคุณหรือภายนอกอาคาร
2. ทอยลูกเต๋า 1 ลูก และให้ผู้เรียนไปตามเส้นทางดังกล่าวซ้ำตามหน้าลูกเต๋า

ทางเลือก:

ให้ผู้เรียนนึกถึงลำดับท่าเต้น เช่น แตะนิ้วเท้า กระโดด ตะขาขึ้นหนึ่งข้าง ทอยลูกเต๋า 1 ลูก และให้ผู้เรียนทำลำดับท่าเต้นซ้ำในจำนวนครั้งตามหน้าลูกเต๋า



เล่น

วัตถุประสงค์: ผู้เขียนจะสามารถสร้างปริศนาที่มีรูปแบบการทำซ้ำ และไขปริศนาร่วมกันเป็นกลุ่ม

การเตรียมตัว: ผู้เรียนจะร่วมงานกันเป็นกลุ่มละ 3 คน ใช้เทปขาวเพื่อสร้างตารางแบบ 4x4 ไว้บนพื้นให้กับแต่ละกลุ่ม

คำแนะนำ:

1. แจกสื่อการเรียน และแบ่งผู้เรียนออกเป็น 3 กลุ่ม
2. อ่านแต่ละบทบาทอย่างละเอียด และกำหนดบทบาทให้กับแต่ละคนในกลุ่มสำหรับเกมแรก
3. ให้ผู้เรียนเล่นเกม โดยเริ่มต้นด้วยบทบาทนักออกแบบ
4. เล่น 3 ครั้ง โดยที่มีการหมุนเวียนการ์ดบทบาทในแต่ละครั้ง

บทบาท:

- นักออกแบบ: ด้วยความช่วยเหลือจากเพื่อนของคุณ วางเพชร 3 เม็ดไว้ในรูปแบบที่ทำซ้ำไว้บนตาราง วางลูกศรเริ่มต้นไว้บนตาราง
- โปรแกรมเมอร์: ด้วยความช่วยเหลือจากเพื่อนของคุณ วางการ์ดคำสั่งไว้บนหรือถัดจากตารางดังกล่าว เพื่อนำทาง Byte ไปยังเพชรและทำการเก็บ ใช้การ์ด Loop เพื่อออกผู้ทดสอบว่าต้องทำการ Loop ผ่านคำสั่งต่างๆ ที่รอบ
- ผู้ทดสอบ: เริ่มต้นโดยให้ Byte อยู่ที่ลูกศร ทำตามการ์ดคำสั่งเพื่อให้ Byte เดินไปตามตาราง หากคุณเก็บเพชรได้ครบแล้ว ก็ฉลองกันได้เลย แต่หากไม่ได้ ให้ร่วมมือกันในทีมเพื่อแก้ไขโค้ด

ทางเลือก:

หากผู้เรียนกำลังทำงานกับคุณคนเดียวหรือเรียนที่บ้าน พวกเขาสามารถเล่นเกมด้วยตัวเองได้โดยใช้กิจกรรม Keynote ทางเลือกที่มีให้ดาวน์โหลด

สื่อสำหรับวิทยากร:

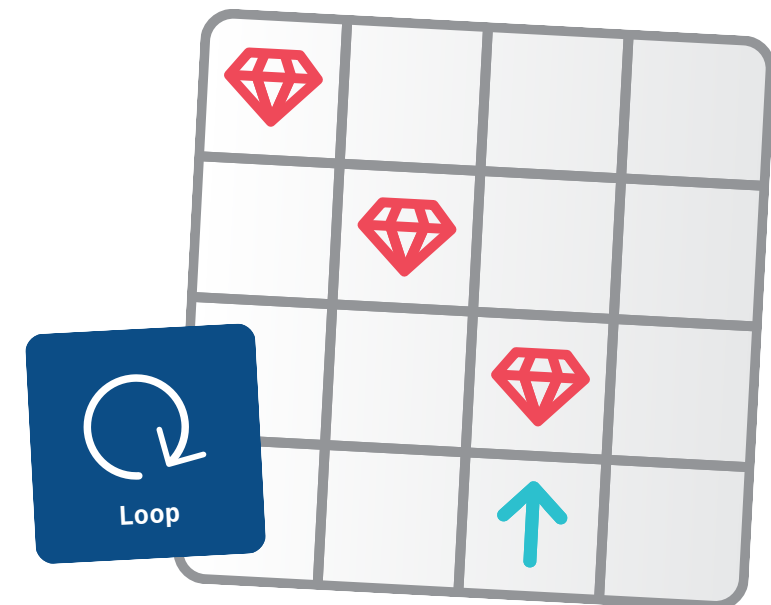
- เทปขาว

สื่อสำหรับผู้เรียน:

- การ์ดบทบาท
- การ์ดคำสั่ง: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` และ `Loop`
- เพชร
- Byte
- ลูกศร

↓ ดาวน์โหลดสื่อ

↓ ดาวน์โหลดกิจกรรมทางเลือก



สำรวจ

วัตถุประสงค์: ศึกษารูปแบบการทำซ้ำในเพลง

การอภิปราย: ให้ผู้เรียนแชร์ข้อมูลเกี่ยวกับเครื่องดนตรีที่ตนเล่นหรือเพลงที่ร้อง ถามว่าพวกเขาเคยเล่นจังหวะหรือร้องท่อนคอรัสซ้ำๆ ตอนเล่นหรือร้องหรือไม่ พวกเขาสามารถนึกถึงส่วนอื่นของดนตรีหรือเพลงที่ซ้ำๆ ได้หรือไม่

สิ่งที่ควรจำ: เสริมแนวคิดที่ว่า Loop ประกอบด้วยสองส่วน:

- คำสั่ง
- จำนวนครั้งที่ทำซ้ำ

ค้นพบ

วัตถุประสงค์: ผู้เรียนจะสามารถทำซ้ำรูปแบบการเล่นกลอง โดยสร้างการเชื่อมโยงระหว่างโค้ดแบบ Loop และตัวอย่างในชีวิตจริง

วัตถุประสงค์:

- สิ่งที่เคยจังหวะได้ เช่น ฟัน ตันขา หรือหนังสือ
- พื้นที่ว่างสำหรับนั่งล้อมกันเป็นวงกลม

คำแนะนำ:

1. ให้ผู้เรียนนั่งล้อมวงกันเป็นวงกลม
2. บอกให้ผู้เรียนเคาะจังหวะกลองที่คุณสร้างไว้ซ้ำๆ กันตามจำนวนนิ้วมือที่คุณชูขึ้นมา เช่น หากคุณชู 4 นิ้ว ผู้เรียนต้องเคาะจังหวะกลองซ้ำ 4 รอบแล้วหยุด
3. ผลัดกันทำไปรอบวงหรือแบ่งออกเป็นกลุ่มเล็กๆ เพื่อให้แต่ละคนมีโอกาสเป็นคนตีกลองนำ

ส่วนเพิ่มเติม:

ให้ผู้เรียนจัดหากลอง



เล่น

วัตถุประสงค์: ผู้เรียนจะเรียกใช้คำสั่งต่างๆ ภายใน Loop และพิจารณาว่าจะต้องเรียกใช้ Loop ดังกล่าวกี่ครั้ง

คำแนะนำ:

1. ฉายรูปสนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1" ขึ้นบนหน้าจอ ไปยังหน้าที่ 3 "ไปจนสุดขอบแล้วกลับมา" ของบท "For Loop"
2. ไปจนสุดขอบแล้วกลับมา:
 - ทบทวนคำสั่ง `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` และ `toggleSwitch()`
 - ให้ผู้เรียนทดลองวิธีการต่างๆ ในการนำทาง Byte จากลูกศรเริ่มต้นไปยังสวิตช์ที่ปิดอยู่แล้วทำการเปิด
 - ในการเพิ่ม For Loop ให้ใช้คำแนะนำโค้ดที่ด้านล่างของตัวแก้ไขหรือ + ที่ด้านบนหน้าจอ
 - รวบรวมไอเดียจากชั้นเรียน และเขียนโค้ดลงในแอป Swift Playgrounds เพื่อไขปริศนาคลิกหรือแตะ "รันโค้ดของฉัน"
 - ลองไอเดียที่แตกต่างกันหลายๆ อย่าง
 - จลองไปกับ Byte
3. ออกจาก "เรียนรู้วิธีเขียนโค้ด 1" และเปิดสนามเด็กเล่น MeeBot Dances แล้วไปที่หน้า Dance Loop (ไม่มีเวิร์กชิตของผู้เรียนสำหรับหน้าสนามเด็กเล่นนี้)
4. Dance Loop:
 - ให้ผู้เรียนใช้ iPad ของตนเองในการทำงานเป็นกลุ่ม คู่ หรือคนเดียวเพื่อทำ Loop แล้วดูการเต้นของหุ่นยนต์
 - ขอให้ผู้เรียนแชร์การเต้นของตัวเอง หรือสร้างการเต้นที่แตกต่างออกไปร่วมกันในชั้นเรียน
 - ออกสแต็ปการเต้นด้วยหุ่นยนต์



เรียนรู้วิธีเขียนโค้ด 1



MeeBot Dances

สื่อสำหรับวิทยากร:

- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 1"
- สนามเด็กเล่น MeeBot Dances
- โปรเจ็กเตอร์หรือจอภาพ

สื่อสำหรับผู้เรียน:

- เวิร์กชิต "ไปจนสุดขอบแล้วกลับมา"
- ดินสอ
- อุปกรณ์ iPad (ไม่บังคับ)
- กระดาษเพิ่มเติม (ไม่บังคับ)



ดาวน์โหลดเวิร์กชิต
"เรียนรู้วิธีเขียนโค้ด"



ตัวแปร



ภาพรวม

บทเรียนที่ 1: ลอยหรือจม

- สำรอง: การอภิปรายเกี่ยวกับการอัปเดตตัวแปร
- ค้นพบ: กิจกรรม "ลอยหรือจม"
- เล่น: เกมการติดตามและตัวอย่าง

บทเรียนที่ 2: เกมคำศัพท์

- สำรอง: การอภิปรายเกี่ยวกับประเภทของคำตอบที่มีต่อคำถาม
- ค้นพบ: กิจกรรมเกมคำศัพท์
- เล่น: เกมปริศนาต่างๆ

บทเรียนที่ 3: ทุกเรื่องเกี่ยวกับตัวฉัน

- สำรอง: การอภิปรายเกี่ยวกับการตอบคำถามด้วยรายการต่างๆ
- ค้นพบ: กิจกรรมทุกเรื่องเกี่ยวกับตัวฉัน
- เล่น: การใช้ Loop

ผู้เรียนจะสามารถ

- เชื่อมโยงชื่อตัวแปรกับค่าที่กำหนด
- เปลี่ยนค่าที่กำหนดให้กับตัวแปร
- ทำความเข้าใจ Swift ประเภทต่างๆ ที่คุณสามารถกำหนดให้กับตัวแปรได้ รวมไปถึงถึงจริง/เท็จ (บูลีน), ตัวเลข (จำนวนเต็ม), คำ (สตริง), สี (Color Literal) และภาพ (Image Literal)
- ทดสอบและดีบักคำสั่งและโค้ด

คำศัพท์

- **ตัวแปร:** คอนเทนเนอร์ที่มีชื่อซึ่งจัดเก็บค่าและทำการเปลี่ยนแปลงได้
- **ข้อมูล:** ข้อมูล
- **บูลีน:** ประเภทที่มีค่าเป็น true หรือ false

มาตรฐาน

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

สำรวจ

วัตถุประสงค์: สำรวจแนวคิดเกี่ยวกับตัวแปร ด้วยการนับสิ่งของและอัปเดตค่าตัวแปร

สื่อสำหรับวิทยากร:

- ไวท์บอร์ด
- ปากกามาร์คเกอร์
- ยางลบ
- กล่อง
- ดินสอ 5 แท่ง (หรือของแบบเดียวกัน 5 ชิ้น)

คำแนะนำ:

1. เริ่มต้นด้วยการเขียนคำสั่งตัวแปรลงบนไวท์บอร์ดเพื่อติดตามสิ่งของของคุณ
 - ตัวอย่าง: `var numberOfPencils = 0`
2. ยกกล่องเปล่าขึ้น และบอกผู้เรียนว่ากล่องดังกล่าวแสดงถึงตัวแปรของคุณ `numberOfPencils`
3. ใส่ดินสอ 1 แท่งลงในกล่อง และถามผู้เรียนว่าค่าตัวแปรตอนนี้เป็นเท่าใด เมื่อพวกเขาตอบถูกต้องแล้ว ให้ลบ 0 ออกและเขียน 1
4. ทำต่อไปจนกว่าคุณใส่ดินสอทั้งหมดลงไปและโค้ดของคุณเป็น:
`var numberOfPencils = 5`
5. จากนั้น ให้เริ่มนำดินสอออกจากกล่อง โดยทำการอัปเดตตัวแปรในขณะที่น่าออก

สิ่งที่ควรจำ: ช่วยให้ผู้เรียนเข้าใจว่าตัวแปรเป็นสิ่งที่เก็บข้อมูลเล็กน้อยๆ ไว้ ในกรณีนี้ ข้อมูลคือตัวเลข และตัวเลขบอกให้คุณรู้ถึงจำนวนดินสอในกล่อง

ค้นพบ

วัตถุประสงค์: เมื่อใช้วัตถุที่พบ ผู้เรียนจะทำการทดลองเพื่อพิจารณาว่าสิ่งของจะลอยหรือจม แล้วพวกเขาจะบันทึกข้อมูลดังกล่าวไว้โดยใช้ภาพ (Image Literal) และค่าจริง/เท็จ (บูลีน)

สื่อสำหรับผู้เรียน:

- iPad
- แอป Keynote
- เวิร์กชีต "ลอยหรือจม"
- ถังน้ำ
- สิ่งของสำหรับทดสอบหลายๆ ชิ้น

คำแนะนำ:

1. แบ่งผู้เรียนออกเป็นกลุ่มเล็กๆ
2. ให้พวกเขารวบรวมสิ่งของหลายๆ อย่างเพื่อมาทดสอบ
3. ในสิ่งของแต่ละชิ้น ให้ผู้เรียน
 - ถ่ายภาพสิ่งของและเพิ่มลงในเวิร์กชีต
 - ทดสอบในน้ำ
 - บันทึกผลการทดสอบลงในเวิร์กชีตด้วยการวงกลมที่ `true` หรือ `false`



ดาวน์โหลดเวิร์กชีต "ลอยหรือจม"

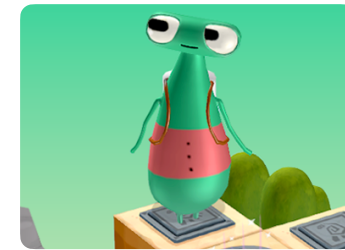
เล่น

วัตถุประสงค์: ผู้เรียนจะสามารถสร้างและอัปเดตตัวแปรต่างๆ ในบริบทด้านการเขียนโค้ดที่แตกต่างกันได้ 2 แบบ

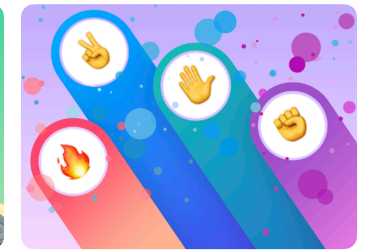
คำแนะนำ:

1. ฉายรูปสนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 2" ขึ้นบนหน้าจอ ไปที่บท "ตัวแปร"
2. บทนำ:
 - อ่านเนื้อหาในหน้าต่างๆ ร่วมกันในชั้น และหยุดถามคำถามหากต้องการ
3. คอยติดตามดู:
 - ให้ผู้เรียนทดลองวิธีการต่างๆ ในการนำทาง Hopper จากลูกศรเริ่มต้นไปยังเพชรและทำการเก็บ ลองบันทึกคำสั่งต่างๆ ไว้ในเวิร์กชีตหรือในแผ่นกระดาษแยกต่างหาก
 - รวบรวมไอเดียจากชั้นเรียน และเขียนโค้ดลงในแอป Swift Playgrounds เพื่อไขปริศนาคลิกหรือแตะ "รันโค้ดของฉัน"
 - ลองไอเดียที่แตกต่างกันหลายๆ อย่าง
 - จลองไปกับ Hopper
4. ออกจาก "เรียนรู้วิธีเขียนโค้ด 2" และไปที่หน้าสุดท้ายของสนามเด็กเล่น "เป่ายิงฉุบ" ที่เรียกว่า "เกมตัวอย่าง" (ไม่มีเวิร์กชีตของผู้เรียนสำหรับหน้าสนามเด็กเล่นนี้)
5. เกมตัวอย่าง:
 - คลิกหรือแตะ "รันโค้ดของฉัน" เพื่อเล่นเกมก่อนที่จะทำการเปลี่ยนแปลงใดๆ
 - พิจารณาร่วมกันเป็นกลุ่มว่าคุณต้องการปรับแต่งส่วนใดของเกม บางส่วนอันน่าสนุกที่คุณสามารถเปลี่ยนแปลงได้ประกอบด้วย `game.roundsToWin`, `game.challenger.emoji`, `game.addOpponent` และ `game.roundPrize`
 - เล่นเกมหลายๆ รอบ โดยเปลี่ยนแปลงบางอย่างให้แตกต่างออกไปทุกครั้ง

ส่วนเพิ่มเติม: สร้างตัวแปรต่างๆ ไว้ในไฟล์ `Game.swift` หากผู้เรียนสงสัยว่าทำไมตัวแปรบางส่วนไม่มี `var` อยู่ด้านหน้า ให้เปิดไฟล์ `Game.swift` เพื่อแสดงตำแหน่งที่สร้างคุณสมบัติของเกม



เรียนรู้วิธีเขียนโค้ด 2



เป่ายิงฉุบ

สื่อสำหรับวิทยากร:

- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น "เรียนรู้วิธีเขียนโค้ด 2"
- สนามเด็กเล่น "เป่ายิงฉุบ"
- โปรเจ็กเตอร์หรือจอภาพ

สื่อสำหรับผู้เรียน:

- เวิร์กชีต "คอยติดตามดู"
- ดินสอ
- กระดาษเพิ่มเติม (ไม่บังคับ)



ดาวน์โหลดเวิร์กชีต
"เรียนรู้วิธีเขียนโค้ด"

สำรวจ

วัตถุประสงค์: ศึกษาคำตอบประเภทต่างๆ ในชีวิตจริงและเชื่อมโยงกับ Swift ประเภทต่างๆ รวมไปถึงใช่/ไม่ใช่ หรือจริง/เท็จ (บูลีน), ตัวเลข (จำนวนเต็ม), คำ (สตริง), สี (Color Literal) และภาพ (Image Literal)

สื่อสำหรับวิทยากร:

- ไวท์บอร์ด
- ปากกามาร์คเกอร์

การอภิปราย: ให้ชั้นเรียนนึกถึงคำถามบางอย่างที่มีคำตอบหลากหลายประเภท และเขียนลงบนกระดาน

ตัวอย่าง:

- ดวงตาของคุณคืออะไร —> สี
- คุณมีสัตว์เลี้ยงหรือไม่ —> ใช่/ไม่
- คุณมีพี่น้องหรือไม่ —> ใช่/ไม่
- คุณอายุเท่าไร —> number
- คุณชื่ออะไร —> คำ

สิ่งที่ควรจำ: อธิบายว่าตัวแปรเองก็มีประเภทที่แตกต่างกัน รวมไปถึงถึงตัวเลข คำ สี ภาพ และคำตอบใช่หรือไม่ คุณจะต้องคงตัวแปรประเภทเดียวกันไว้ แม้ว่า คุณจะอัปเดตตัวแปรเป็นแบบใหม่ก็ตาม ทั้งนี้ขึ้นอยู่กับวิธีการสร้างตัวแปร เช่น `var myAge = 8` สามารถเปลี่ยนเป็น 9 ได้ แต่เปลี่ยนเป็น "nine" ไม่ได้

ค้นพบ

วัตถุประสงค์: ผู้เรียนจะสามารถเล่นเกมคำศัพท์ด้วยการกรอกประเภทคำตอบที่ถูกต้อง

สื่อสำหรับผู้เรียน:

- เวอร์กซ์เกมคำศัพท์
- ดินสอ
- ดินสอสี

คำแนะนำ:

ให้ผู้เรียนเล่นเกมคำศัพท์ร่วมกันเป็นกลุ่มเล็กๆ อย่างน้อย 1 เกม แต่ละกลุ่มควรมีคนอ่านหรือคนคอยสนับสนุนอย่างน้อย 1 คน หากผู้เรียนทุกคนไม่สามารถอ่านได้ ให้เล่นเกมเล็กๆ ร่วมกันในกลุ่มใหญ่

ส่วนเพิ่มเติม: หากผู้เรียนทำได้ ให้พวกเขาคิดเกมคำศัพท์ให้คู่ของตนทำ แนะนำให้พวกเขาใช้ตัวเลข คำศัพท์ สี ภาพ และคำถามแบบใช่หรือไม่ในส่วนของช่องว่าง



[ดาวน์โหลดเวิร์กซ์เกมคำศัพท์](#)

เล่น

วัตถุประสงค์: ผู้เรียนจะสามารถนำทาง Byte ไปเก็บเพชร และเพิ่มเพชรแต่ละเม็ดลงในกล่อง และอัปเดตค่าตัวแปร

การเตรียมตัว: ผู้เรียนจะร่วมงานกันเป็นกลุ่มละ 3 คน ใช้เทปขาวเพื่อสร้างตารางแบบ 4x4 ไว้บนพื้นให้กับแต่ละกลุ่ม

คำแนะนำ:

1. แจกสื่อการเรียนรู้ และแบ่งผู้เรียนออกเป็น 3 กลุ่ม
2. อ่านแต่ละบทบาทอย่างละเอียด และกำหนดบทบาทให้กับแต่ละคนในกลุ่มสำหรับเกมแรก
3. ให้ผู้เรียนเล่นเกม โดยเริ่มต้นด้วยบทบาทนักออกแบบ
4. เล่น 3 ครั้ง โดยที่มีการหมุนเวียนการ์ดบทบาทในแต่ละครั้ง

บทบาท:

- นักออกแบบ: วางเพชรหลายๆ เม็ดและลูกศรเริ่มต้นไว้บนตาราง
- โปรแกรมเมอร์: ด้วยความช่วยเหลือจากเพื่อนของคุณ วางการ์ดคำสั่งไว้บนหรือถัดจากตารางดังกล่าว เพื่อนำทาง Byte ไปยังเพชรและทำการเก็บ
- ผู้ทดสอบ: เริ่มต้นโดยให้ Byte อยู่ที่ลูกศร ทำตามคำสั่งเพื่อให้ Byte เดินไปตามตาราง โดยใส่เพชรเพิ่มลงในกล่องเมื่อเก็บรวบรวมได้ หากคุณเก็บเพชรได้ครบแล้ว ให้อัปเดตตัวแปร `numberOfGems` ในกล่อง แล้วก็ลองกันได้เลย แต่หากเก็บได้ไม่ครบ ให้ร่วมมือกันในทีมเพื่อแก้ไขโค้ด

ทางเลือก:

หากผู้เรียนกำลังทำงานกับคุณคนเดียวหรือเรียนที่บ้าน พวกเขาสามารถเล่นเกมด้วยตัวเองได้โดยใช้กิจกรรม Keynote ทางเลือกที่มีให้ดาวน์โหลด

สื่อสำหรับวิทยากร:

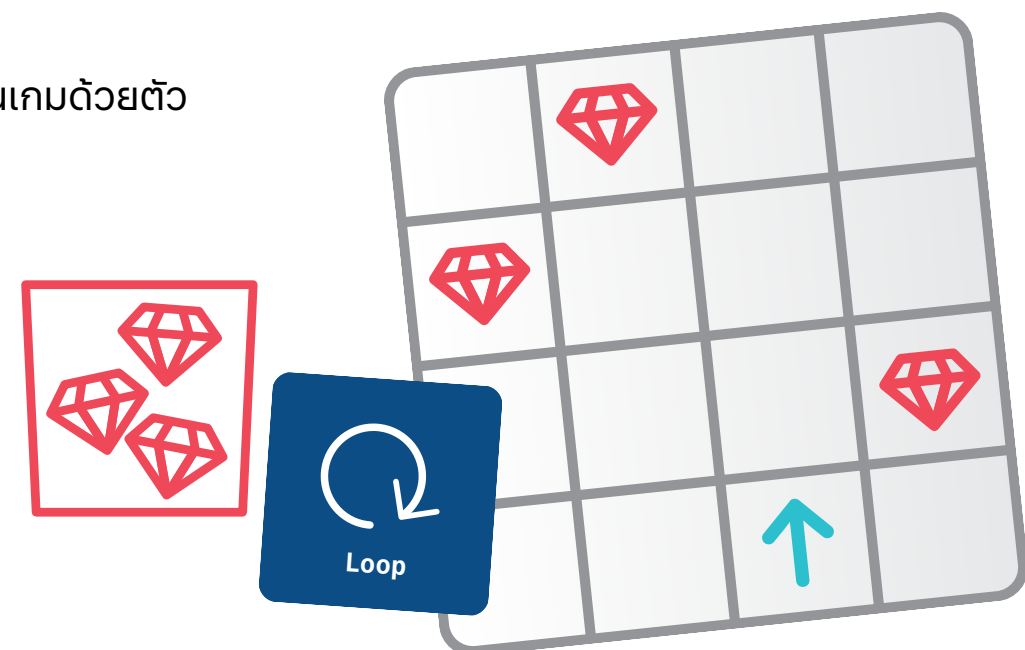
- เทปขาว

สื่อสำหรับผู้เรียน:

- การ์ดบทบาท
- การ์ดคำสั่ง: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` และ `Loop`
- เพชร
- Byte
- ลูกศร
- กล่องที่ติดป้าย: `var numberOfGems = _____`
- ปากกา

↓ [ดาวน์โหลดสื่อ](#)

↓ [ดาวน์โหลดกิจกรรมทางเลือก](#)












สำรวจ

วัตถุประสงค์: สำรวจแนวทางการใช้รายการหรืออาร์เรย์เมื่อทำการสร้างตัวแปร

การอภิปราย: จะเป็นอย่างไร หากเวิร์กชิตต้องการชื่อพี่น้องของผู้เรียนและเจ้าตัวมีพี่น้องมากกว่าหนึ่งคน รวบรวมไอเดียจากชั้นเรียน หากพวกเขาแนะนำให้ทำรายการ ให้บอกพวกเขาถึงสิ่งที่คนเขียนโค้ดทำ ในกรณีที่ตัวแปรมีมากกว่า 1 คำตอบ ผู้เรียนควรสร้างรายการขึ้นมา

ให้ผู้เรียนนึกถึงคำถามที่อาจมีหลายคำตอบ

ตัวอย่าง:

- ชื่อของเพื่อน —> Rose, Sam, Joy
- อายุของผู้เรียน —> 7, 8, 7, 8, 7, 8, 9, 7, 8, 9, 8
- สีโปรด —>  ,  ,  ,  , 
- สัตว์ที่ชอบ —>  ,  ,  , 

สิ่งที่ควรจำ: รายการที่ผู้เรียนสร้างไว้ในโค้ดเป็นเหมือนกับรายการประโยคต่างๆ

ค้นพบ

วัตถุประสงค์: ผู้เรียนจะสามารถใส่ตัวแปรต่างๆ เพื่ออธิบายถึงสิ่งต่างๆ เกี่ยวกับตัวเองและคู่ของตน ผู้เรียนอาจมีโอกาใช้อาร์เรย์เป็นประเภทตัวแปรได้

สื่อสำหรับผู้เรียน:

- เวิร์กชิต "ทุกเรื่องเกี่ยวกับตัวฉัน" และ "ทุกเรื่องเกี่ยวกับตัวคุณ"
- ดินสอ
- ดินสอสี

คำแนะนำ:

1. ให้ผู้เรียนทำเวิร์กชิต "ทุกเรื่องเกี่ยวกับตัวฉัน" ให้เสร็จสิ้น
 - หากผู้เรียนมีพี่น้องหรือสัตว์เลี้ยงมากกว่าหนึ่ง ให้พวกเขาทำรายการคั่นด้วยเครื่องหมายจุลภาค
2. จับคู่ผู้เรียนให้ทำเวิร์กชิต "ทุกเรื่องเกี่ยวกับตัวฉัน" ให้เสร็จสิ้น

ทางเลือก: ผู้เรียนสามารถใช้ iPad และ Keynote ในการทำเวิร์กชิตถ่ายภาพเพื่อเป็นรูปภาพสำหรับคำตอบ และใส่สีให้กับ Color Literal โดยใช้ตัวเลือกในการจัดรูปแบบ



ดาวน์โหลดเวิร์กชิต "ทุกเรื่องเกี่ยวกับ"

เล่น

วัตถุประสงค์: ผู้เรียนจะสามารถระบุตัวแปรในโค้ดและศึกษาแนวทางที่ตนสามารถใช้อาร์เรย์ใน Loop ได้

คำแนะนำ:

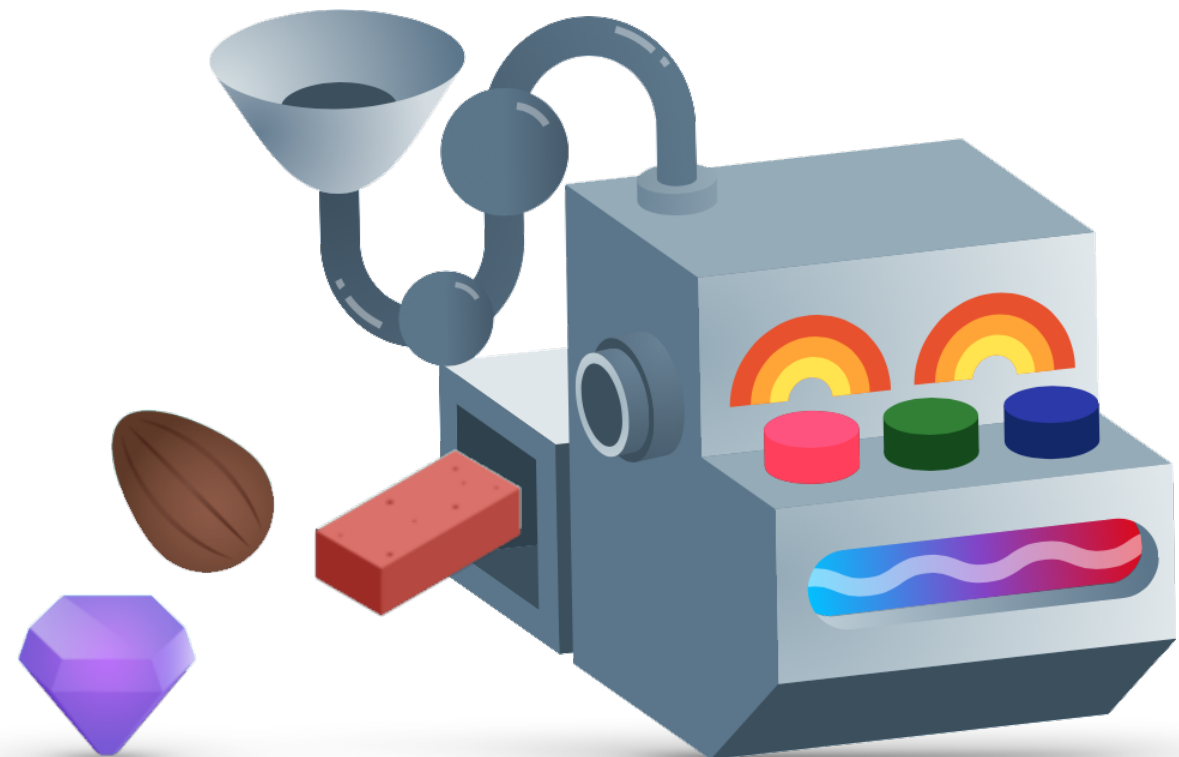
1. ฉายรูปสนามเด็กเล่น "เครื่องเขียนโค้ด" ขึ้นบนหน้าจอ
2. บทนำ:
 - อ่านเนื้อหาในหน้าต่างๆ ร่วมกันในชั้น และหยุดถามคำถาม หากต้องการ
 - ไม่บังคับ: เล่นผ่าน 2 หน้าแรก "การสำรวจเครื่องจักร" และ "การผสมผสานด้วยสี"
3. การใช้ Loop:
 - ในหน้านี้ ผู้เรียนจะได้ผสมผสานความรู้ด้าน Loop เข้ากับตัวแปรต่างๆ
 - ดูว่าผู้เรียนสามารถระบุตัวแปรต่างๆ ในโค้ดที่ใช้อาร์เรย์ได้หรือไม่
 - คลิกหรือแตะ "รันโค้ดของฉัน" เพื่อดูสิ่งที่เครื่องสร้างขึ้น
 - ไปที่ขั้นตอนที่ 2 ในบทนำ และอัปเดตโค้ดให้รวมถึงตัวแปรที่สอง สิ่งของ และ Loop ที่ซ้อนกัน คลิกหรือแตะ "รันโค้ดของฉัน" อีกครั้งเพื่อดูสิ่งที่เครื่องสร้างขึ้น
 - หมายเหตุ: ลองทำหน้านี้ก่อนทำกิจกรรมกับผู้เรียน



เครื่องเขียนโค้ด

สื่อสำหรับวิทยากร:

- iPad หรือ Mac
- แอป Swift Playgrounds
- สนามเด็กเล่น "เครื่องเขียนโค้ด"
- โปรเจ็กเตอร์หรือจอภาพ



การออกแบบแอป



สำรวจ

วัตถุประสงค์: สำรวจแอปที่คุณเคยในอุปกรณ์ต่างๆ

คำแนะนำ: เริ่มการอภิปรายเกี่ยวกับแอปที่ผู้เรียนใช้ใน iPad ที่บ้านหรือโรงเรียน แล้วพูดถึงแอปที่พวกเขาหรือผู้ปกครองใช้ในอุปกรณ์ที่บ้าน

สิ่งที่ควรจำ: เสริมแนวคิดที่ว่าแอปไม่ได้มีเพียงแต่ใน iPhone เท่านั้น แต่ยังมีในนาฬิกา แท็บเล็ต คอมพิวเตอร์ และทีวีอีกด้วย

ส่วนเพิ่มเติม: เจาะลึกถึงตัวอย่างแอปบางส่วน พร้อมกับถามผู้เรียนว่าแอปดังกล่าวออกแบบมาเพื่อใคร ทำอะไรได้บ้าง และจุดประสงค์ในการทำแอปนั้นขึ้นมา

ตัวอย่าง:

- แอป: Swift Playgrounds
- ทำมาเพื่อใคร: คนที่ต้องการเรียนรู้เกี่ยวกับ Swift
- ทำอะไรได้: ช่วยผู้คนในการเรียนรู้วิธีการเขียนโค้ดผ่านปริศนาและบทเรียนต่างๆ
- จุดประสงค์ในการทำแอป: เพื่อสอนวิธีการเขียนโค้ดให้กับผู้ที่มีความรู้ด้านการเขียนโปรแกรมเล็กน้อยหรือไม่มีเลย

ค้นพบ

วัตถุประสงค์: เตรียมผู้เรียนให้พร้อมสำหรับการออกแบบแอปของตัวเองด้วยการวิเคราะห์แอปที่คุณเคย

สื่อสำหรับผู้เรียน:

- iPad
- เวิร์กชิต "แอปคืออะไร"
- ดินสอ
- ปากกาหรือดินสอสี

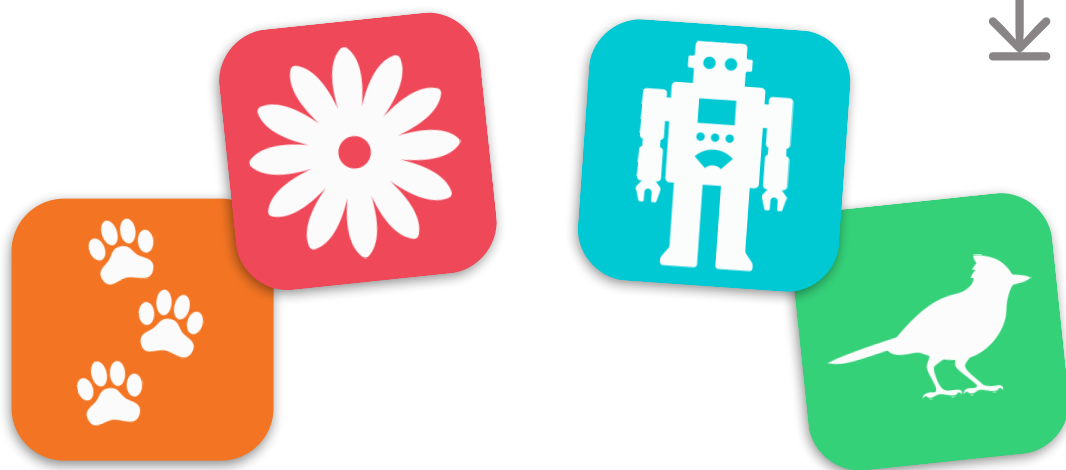
คำแนะนำ:

1. แบ่งผู้เรียนออกเป็นกลุ่มเล็กๆ หรือให้พวกเขาทำคนเดียว
2. ให้ผู้เรียนเลือกแอป iPad
3. ให้พวกเขาใช้เวิร์กชิต "แอปคืออะไร" เพื่อเป็นแนวทางในการสำรวจแอปของตน
4. ชวนให้ผู้เรียนแชร์สิ่งที่ตัวเองพบเกี่ยวกับแอปดังกล่าวร่วมกับกลุ่มใหญ่หรือคู่ของตน

เคล็ดลับสำหรับวิทยากร: ยิ่งผู้เรียนมีอายุน้อยเท่าใด พวกเขายิ่งต้องการความช่วยเหลือในการทำเวิร์กชิตนี้มากเท่านั้น สำหรับกลุ่มอนุบาลศึกษาปีที่ 1 ให้ลองทำ 2 หรือ 3 แอปเป็นกลุ่มใหญ่



ดาวน์โหลดเวิร์กชิต "แอปคืออะไร"



เล่น

วัตถุประสงค์: ผู้เรียนออกแบบแอปของตัวเองได้

สื่อสำหรับผู้เรียน:

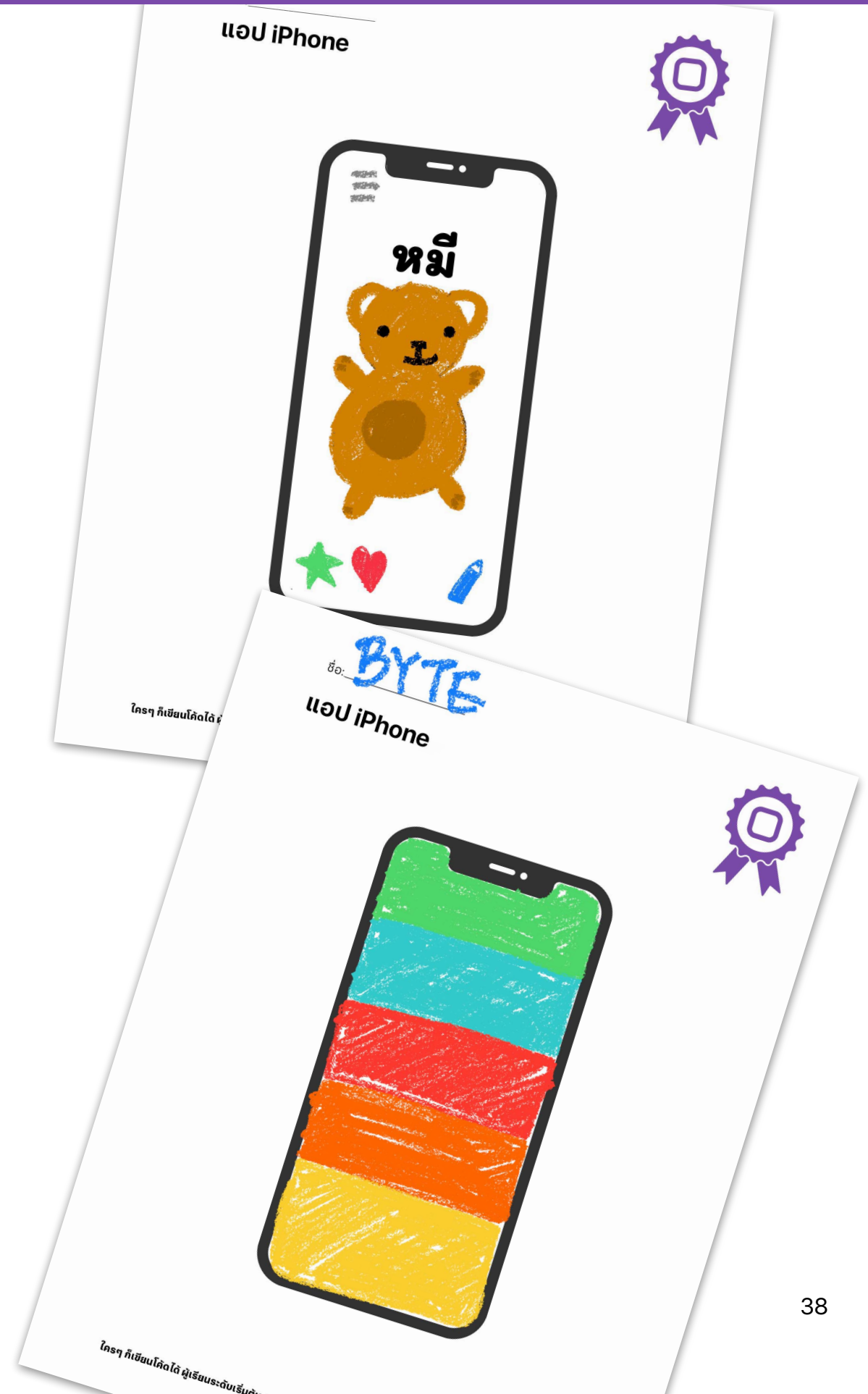
- เวอร์กชิตการออกแบบแอปของฉันทัน
- เกมเพลตอุปกรณ์
- กระดาษเพิ่มเติม
- ดินสอ
- ปากกาหรือดินสอสี

คำแนะนำ:

1. แบ่งผู้เรียนออกเป็นกลุ่มเล็กๆ หรือให้พวกเขาทำคนเดียว
2. อธิบายเวอร์กชิตการออกแบบแอปของฉันทันกับผู้เรียนเพื่อแนะนำกระบวนการขั้นตอนของพวกเขาในการออกแบบแอป
3. ให้ผู้เรียนสร้างต้นแบบหน้าแอปของตนเองโดยใช้กระดาษเพิ่มเติมหรือเกมเพลตอุปกรณ์
4. แนะนำให้ผู้เรียนสร้างต้นแบบของแอปเวอร์ชันสุดท้ายโดยใช้เกมเพลตอุปกรณ์
5. เชิญผู้เรียนแต่ละคนหรือกลุ่มผู้เรียนแต่ละกลุ่มมานำเสนอไอเดียของตนกับกลุ่มใหญ่

↓ [ดาวน์โหลดเวอร์กชิตการออกแบบแอปของฉันทัน](#)

↓ [ดาวน์โหลดเกมเพลตอุปกรณ์](#)



แหล่งข้อมูลวิทยากร



อภิธานศัพท์

- **บูลีน:** ประเภทที่มีค่าเป็น true หรือ false
- **บ๊ิต:** ข้อผิดพลาดในโค้ด
- **คำสั่ง:** โค้ดที่สั่งให้แอปพลิเคชันดำเนินการอย่างไรอย่างหนึ่ง
- **ข้อมูล:** ข้อมูล
- **ดีบัก:** การค้นหาและแก้ไขข้อผิดพลาดในโค้ด
- **ฟังก์ชัน:** ชุดคำสั่งที่มีชื่อ ซึ่งสามารถนำไปใช้ได้เมื่อต้องการ
- **Loop:** โค้ดชุดหนึ่งที่จะรันซ้ำๆ ตามจำนวนครั้งที่กำหนด
- **ปรับเปลี่ยน:** การเปลี่ยนแปลง
- **ลำดับ:** ลำดับของสิ่งที่เกิดขึ้น
- **ขั้นตอน:** การดำเนินการหนึ่งอย่างในกระบวนการที่ใหญ่กว่าปกติ
- **สลับเปิดปิด:** เปิดหรือปิด
- **ตัวแปร:** คอนเทนเนอร์ที่มีชื่อซึ่งจัดเก็บค่าและทำการเปลี่ยนแปลงได้

มาตรฐาน CSTA >

1A-AP

- 1A-AP-08: จำลองกระบวนการในแต่ละวันด้วยการสร้างและทำตามอัลกอริทึม (ชุดคำสั่งแบบทีละขั้นตอน) ในการทำงานให้เสร็จสิ้น
- 1A-AP-09: จำลองวิธีการที่โปรแกรมจัดเก็บและจัดการกับข้อมูลโดยใช้ตัวเลขหรือสัญลักษณ์อื่นๆ เพื่อแสดงถึงข้อมูล
- 1A-AP-10: พัฒนาโปรแกรมที่มีลำดับและ Loop อย่างง่ายๆ ในการถ่ายทอดโอเดียหรือแก้ไขปัญหา
- 1A-AP-11: แยก (แบ่งออกเป็นส่วนๆ) ขั้นตอนต่างๆ ที่จำเป็นสำหรับการแก้ไขปัญหาลำดับคำสั่งที่ต้องแม่นยำ
- 1A-AP-12: สร้างแผนที่อธิบายถึงลำดับเหตุการณ์ เป้าหมาย และผลลัพธ์ที่คาดหวังของโปรแกรม
- 1A-AP-14: แก้ไข (ระบุและแก้ไข) ข้อผิดพลาดในอัลกอริทึมหรือโปรแกรมที่ประกอบด้วยลำดับและ Loop อย่างง่ายๆ

1A-CS

- 1A-CS-01: เลือกและใช้งานซอฟต์แวร์ที่เหมาะสมในการทำงานต่างๆ และเข้าใจว่าผู้ใช้งานมีความต้องการและความชอบในเทคโนโลยีที่พวกเขาใช้แตกต่างกัน

1B-AP

- 1B-AP-09: สร้างโปรแกรมที่ใช้ตัวแปรต่างๆ ในการจัดเก็บและปรับเปลี่ยนค่า
- 1B-AP-10: สร้างโปรแกรมที่ประกอบด้วยลำดับ, เหตุการณ์, Loop และเงื่อนไข
- 1B-AP-16: สวมบทบาทที่แตกต่างกันไปผ่านคำแนะนำของวิทยากรเมื่อทำงานร่วมกับเพื่อนๆ ในระหว่างขั้นตอนการออกแบบ การปรับใช้ และการตรวจสอบการพัฒนาโปรแกรม

ตัวอย่างคำตอบ

หน้าถัดๆ ไปเหล่านี้ให้ตัวอย่างวิธีแก้ปัญหา 1 อย่างสำหรับปริศนา Swift Playgrounds แต่ละข้อ แต่เราสามารถไขปริศนาต่างๆ ได้มากกว่า 1 วิธี กระตุ้นให้ผู้เรียนลองวิธีการใหม่ๆ ที่พวกเขาสามารถนำทาง Byte หรือตัวละครอื่นๆ ได้

เฉลิมฉลองให้กับการเขียนโค้ดทุกประเภทและเป้าหมายที่ผู้เรียนอาจมี ผู้เรียนบางคนอาจต้องการสำรวจปริศนาทั้งหมดนอกเหนือไปจากการเก็บรวบรวมเพชร ในขณะที่คนอื่นๆ อาจต้องการพลิกแพลงการเก็บรวบรวมเพชรด้วยวิธีการอื่นๆ ให้มากที่สุดเท่าที่ทำได้ อย่าลืมว่า การเขียนโค้ดควรเป็นเรื่องสนุก



เรียนรู้วิธีเขียนโค้ด 1

บท "คำสั่ง"

การออกคำสั่ง

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

บท "คำสั่ง"

การเพิ่มคำสั่งใหม่

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

บท "ฟังก์ชัน"

การเขียนลักษณะการทำงานใหม่

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

บท "ฟังก์ชัน"

การสร้างฟังก์ชันใหม่

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

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



เรียนรู้วิธีเขียนโค้ด 1

บท "ฟังก์ชัน"

เก็บ สลับเปิดปิด ทำซ้ำ

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

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

un Loop

การใช้ Loop

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

un Loop

การใช้ Loop ทุกด้าน

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

un Loop

ไปจนสุดขอบแล้วกลับมา

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

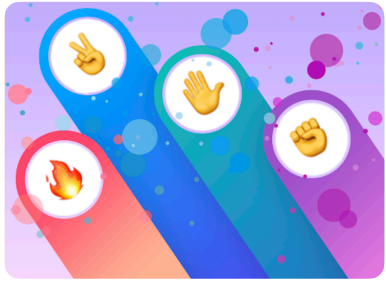


เรียนรู้วิธีเขียนโค้ด 2

บท "ตัวแปร"

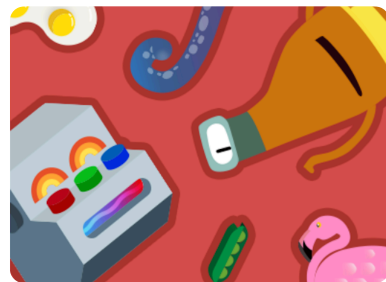
คอยติดตามดู

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

เป่ายิ้งฉุบ เกมตัวอย่าง

ไม่มีตัวอย่างวิธีแก้ปัญห
สำหรับหน้านี้
เนื่องจากเกมเป็นแบบที่
ปรับแต่งได้เต็มรูปแบบ คุณ
จึงสามารถเล่นได้
อย่างที่คุณต้องการ

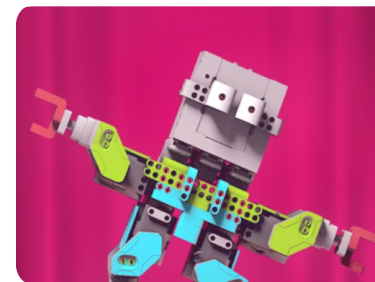


เครื่องเขียนโค้ด การใช้ Loop

```
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()
}
```



MeeBot Dances ท่าเต้นพื้นฐาน

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

Dance Loop

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



© 2022 Apple Inc. สงวนลิขสิทธิ์ทุกประการ Apple, โลโก้ Apple, Apple Watch, iPad, iPhone, Keynote, Mac, Pages, Swift, โลโก้ Swift และ Swift Playgrounds เป็นเครื่องหมายการค้าของ Apple Inc. ซึ่งจดทะเบียนในสหรัฐอเมริกาและประเทศอื่นๆ App Store และ "ใครๆ ก็เขียนโค้ดได้" เป็นเครื่องหมายบริการของ Apple Inc. ซึ่งจดทะเบียนในสหรัฐอเมริกาและประเทศอื่นๆ ชื่อสินค้าและบริษัทอื่นๆ ที่กล่าวถึง ณ ที่นี้อาจเป็นเครื่องหมายการค้าของบริษัทที่เกี่ยวข้อง พฤศจิกายน 2022