



พัฒนาใน Swift

Swift Coding Club



ยินดีต้อนรับสู่ Swift Coding Club

การเรียนรู้เขียนโค้ดจะสอนให้คุณรู้จักวิธีแก้ปัญหาและทำงานร่วมกับผู้อื่นได้อย่างสร้างสรรค์ แลกช่วยเนรมิตไอเดียของคุณให้กลายเป็นจริงได้

Swift Coding Club คือวิธีเรียนเขียนโค้ดและออกแบบแอปที่สนุกสนาน เพราะมีกิจกรรมต่างๆ ที่สร้างขึ้นเพื่อ Swift ซึ่งเป็นภาษาสำหรับการเขียนโค้ดของ Apple ที่จะช่วยส่งเสริมการทำงานร่วมกันในขณะที่คุณเรียนเขียนโค้ด สร้างแอปต้นแบบ และคิดว่าโค้ดจะสร้างความแตกต่างให้กับโลกรอบๆ ตัวคุณได้อย่างไร

คุณไม่จำเป็นต้องเป็นนักการศึกษาหรือโปรแกรมเมอร์ถึงจะสามารถดำเนินกิจกรรมชมรม Swift Coding Club ได้ เพราะคุณสามารถศึกษาเนื้อหาต่างๆ ด้วยตัวเองเมื่อไหร่ก็ได้ หรือจะเรียนไปพร้อมกับสมาชิกชมรมคนอื่นก็ได้เช่นกัน และยังสามารถร่วมกันเฉลิมฉลองไอเดียและการออกแบบของชมรม ด้วยการจัดกิจกรรมการแสดงผลงานแอปในชุมชนของคุณได้ด้วย

คู่มือนี้แบ่งออกเป็น 3 ส่วนดังนี้



เริ่มเลย

ทุกสิ่งที่คุณจำเป็นต้องใช้ในการตั้งชมรม Swift Coding Club



เรียนรู้และนำไปใช้

โมดูลและกิจกรรมต่างๆ สำหรับเซสชันชมรม



เฉลิมฉลอง

แหล่งข้อมูลที่เป็นประโยชน์สำหรับการวางแผนและจัดกิจกรรมในชุมชน

แหล่งข้อมูลการเขียนโค้ด

Swift Coding Club ออกแบบมาโดยใช้แหล่งข้อมูลที่หลากหลายสำหรับการสอนเขียนโค้ด Apple ฟานักเขียนโค้ดก้าวจากการเรียนรู้ขั้นพื้นฐานบน iPad มาสู่การสร้างแอปที่ใช้งานได้จริงบน Mac



ใครๆ ก็เขียนโค้ดได้ | อายุ 10 ปีขึ้นไป

ใช้โค้ด Swift เพื่อเรียนรู้พื้นฐานการเขียนโค้ดด้วย Swift Playgrounds บน iPad เรียนรู้เพิ่มเติมเกี่ยวกับ [หลักสูตร "ใครๆ ก็เขียนโค้ดได้" >](#)



พัฒนาใน Swift | อายุ 14 ปีขึ้นไป

เรียนรู้วิธีพัฒนาแอปใน Xcode บน Mac เรียนรู้เพิ่มเติมเกี่ยวกับ [ข้อเสนอ "พัฒนาใน Swift" >](#)

เรียนรู้การเขียนโค้ดกับ Apple

คุณไม่จำเป็นต้องมีประสบการณ์เบื้องต้นก่อนที่จะศึกษาการสร้างแอปสำหรับแพลตฟอร์ม Apple เพราะหลักสูตรการพัฒนาแอปของ Apple ออกแบบมาให้ง่ายสำหรับทุกคนในการเขียนโค้ด Swift แบบเดียวกับมืออาชีพ ไม่ว่าจะใช้ในภาคการศึกษา ใช้เพื่อการรับรองระดับมืออาชีพ หรือเพื่อต่อยอดทักษะของคุณ [เรียนรู้เพิ่มเติม >](#)

เริ่มเลย

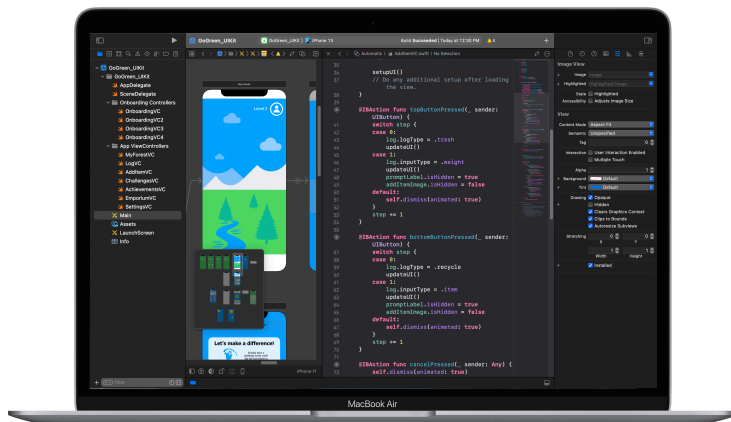


1. ศึกษาแหล่งข้อมูล "พัฒนาใน Swift"

แหล่งข้อมูล "พัฒนาใน Swift" สอนคุณให้ใช้ Xcode บน Mac และ Swift ซึ่งเป็นภาษาการเขียนโปรแกรมแบบโอเพ่นซอร์สที่ทรงพลังและใช้งานง่ายที่ออกแบบโดย Apple ซึ่งเป็นภาษาเดียวกับที่นักพัฒนามืออาชีพใช้เพื่อสร้างแอปสำหรับ iOS, macOS, tvOS, iPadOS, watchOS และอื่นๆ ในแวดวงธุรกิจแอปที่กำลังเติบโตอย่างรวดเร็ว แหล่งข้อมูลนี้จะช่วยสร้างแรงบันดาลใจให้สมาชิกชมรมที่ยังใหม่กับการเขียนโค้ดและที่มีประสบการณ์แล้ว ก่อนที่คุณจะเริ่มออกแบบประสบการณ์ชมรม ให้ศึกษาแหล่งข้อมูล "พัฒนาใน Swift" เหล่านี้

Xcode

Xcode เป็นสภาพแวดล้อมการพัฒนาแบบบูรณาการที่นักพัฒนามืออาชีพใช้สร้างแอปจริง โดยจะมอบเครื่องมือที่คุณต้องใช้สร้างแอปที่สมบูรณ์ ตั้งแต่การออกแบบอินเทอร์เฟซผู้ใช้ (UI) และการใช้โค้ดเพื่อการทดสอบและดีบักแอปของคุณ รวมถึงเตรียมแอปเพื่อเปิดให้ดาวน์โหลดบน App Store



[ดาวน์โหลด Xcode >](#)

พัฒนาใน Swift: การสำรวจ

เรียนรู้แนวคิดการเขียนโปรแกรมคอมพิวเตอร์ที่สำคัญไปพร้อมกับการสร้างรากฐานอันมั่นคงในการเขียนโปรแกรมด้วย Swift พวกเขาจะได้เรียนรู้เกี่ยวกับผลที่การเขียนโปรแกรมคอมพิวเตอร์และแอปมีต่อสังคม เศรษฐกิจ และวัฒนธรรมไปพร้อมๆ กับการศึกษาเกี่ยวกับการพัฒนาแอป iOS บทเรียนจะแนะนำคุณเกี่ยวกับกระบวนการออกแบบแอป ซึ่งประกอบด้วยการระดมความคิด การวางแผน การสร้างต้นแบบ และการประเมินแอปของตนเอง



[ดาวน์โหลด "พัฒนาใน Swift: การสำรวจ" >](#)



[ดาวน์โหลดคู่มือผู้สอนสำหรับ "พัฒนาใน Swift: การสำรวจ" >](#)



2. ตรวจสอบความพร้อมด้านเทคโนโลยี

ก่อนการเจอกันครั้งแรก คุณต้องแน่ใจว่ามีสิ่งต่อไปนี้

- **Mac** คุณต้องมี Mac ที่ใช้ macOS Big Sur หรือใหม่กว่า และแนะนำว่าสมาชิกชมรมแต่ละคนควรมี Mac ของตัวเอง แต่คุณก็ยังสามารถแชร์อุปกรณ์และเขียนโค้ดด้วยกันได้
- **Xcode 12** แอป Mac ฟรีจาก Apple นี้ใช้เพื่อสร้างแอป Mac และแอป iOS ทุกแอป Xcode มีเครื่องมือทั้งหมดที่จำเป็นในการสร้างแอปที่ยอดเยี่ยม
- **"พัฒนาใน Swift: การสำรวจ"** แหล่งข้อมูลฟรีนี้จาก Apple จะช่วยแนะนำนักเขียนโค้ดมือใหม่ให้รู้จักแนวคิดการเขียนโปรแกรมคอมพิวเตอร์ที่สำคัญเพื่อสร้างรากฐานอันมั่นคงในการเขียนโปรแกรมด้วย Swift
- **Keynote** คุณจะใช้แอปนี้บน Mac เพื่อออกแบบต้นแบบแอปของคุณ

ไปที่ [บริการช่วยเหลือของ Apple](#) เพื่อขอรับความช่วยเหลือเกี่ยวกับผลิตภัณฑ์ของ Apple

3. จัดทำแผน

เรื่องที่ต้องพิจารณามีดังนี้

- สมาชิกชมรมมีใครบ้าง แต่ละคนสนใจเรื่องอะไรบ้าง และแต่ละคนมีประสบการณ์ในการเขียนโค้ดมาก่อนหรือไม่ หรือเป็นมือใหม่เลย
- ชมรมจะนัดประชุมกันบ่อยแค่ไหน คุณจะจัดกิจกรรมการเขียนโค้ดนานกี่ชั่วโมง
- ชมรมสามารถเข้าถึงเทคโนโลยีอะไรได้บ้าง
- เป้าหมายของชมรมคืออะไร

4. บอกต่อ

บอกให้ทุกคนรู้เกี่ยวกับชมรม Swift Coding Club ลองดูไอเดียและสื่อต่างๆ สำหรับเชิญชวนสมาชิกใหม่ให้มาเข้าร่วมชมรมของคุณ

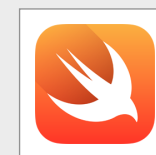
- **ประกาศเกี่ยวกับชมรมของคุณ** ใช้อีเมล โซเชียลมีเดีย เว็บไซต์ ใบปลิว หรือวิธีบอกแบบปากต่อปาก เพื่อให้ชุมชนของคุณรับรู้เกี่ยวกับชมรมนี้
- **จัดการประชุมเพื่อให้ข้อมูล** ถามผู้ที่น่าจะเข้าเป็นสมาชิกชมรมว่าสนใจเรื่องอะไรบ้าง และอยากสร้างโปรเจกต์แบบไหน พูดคุยเกี่ยวกับไอเดียการจัดกิจกรรมในชุมชน และวิธีที่พวกเขาสามารถมีส่วนร่วมได้ คุณยังสามารถแชร์วิดีโอสั้นๆ เกี่ยวกับชมรมทางออนไลน์หรือบนโซเชียลมีเดีย

สื่อต่อไปนี้จะช่วยโปรโมทและปรับแต่ง Swift Coding Club ในแบบที่คุณต้องการได้

- **โปสเตอร์ ดาวน์โหลดเทมเพลตนี้ได้ฟรี** แล้วปรับแต่งโปสเตอร์ในแบบของคุณเอง จากนั้นพิมพ์และนำไปติด หรือจะสร้างโปสเตอร์แบบดิจิทัลแล้วแชร์ทางออนไลน์ก็ได้ และอย่าลืมระบุรายละเอียดวันเวลาและสถานที่นัดหมายของชมรม รวมถึงวิธีเข้าร่วมด้วย
- **สติ๊กเกอร์และเสื้อยืด** ใช้ **สติ๊กเกอร์ชมรม Swift Coding Club เหล่านี้** เพื่อช่วยโปรโมทชมรมเสื้อยืดเหมาะสำหรับมอบให้สมาชิกที่เข้าร่วมงานจัดแสดงแอป ดาวน์โหลด **เทมเพลตเสื้อยืดชมรม Swift Coding Club** เพื่อทำเสื้อยืดแจกให้สมาชิก



โปสเตอร์ชมรม Swift Coding Club



สติ๊กเกอร์ชมรม Swift Coding Club



เสื้อยืดชมรม Swift Coding Club

เคล็ดลับสำหรับหัวหน้าชมรม



สร้างทีมผู้นำ การมีสมาชิกกลุ่มหนึ่งคอยเป็นผู้นำในการทำกิจกรรมต่างๆ ในชมรมจะช่วยให้ทุกอย่างง่ายขึ้น แล้วยังสนุกขึ้นด้วย ลองดูว่าสมาชิกชมรมคนไหนมีแววเป็นหัวหน้าได้บ้าง และคุณอาจพิจารณาเพิ่มเจ้าหน้าที่ประจำชมรมสำหรับการจัดงาน การเขียนโค้ด การออกแบบแอป และอื่นๆ ด้วยก็ได้

เรียนรู้ไปด้วยกัน หัวหน้าชมรมไม่จำเป็นต้องรู้ทุกเรื่อง เพราะหน้าที่ของพวกเขาคือช่วยสมาชิกศึกษาค้นคว้าด้วยตัวเองและพัฒนาทักษะการแก้ไขปัญหารวมถึงสนับสนุนให้สมาชิกช่วยเหลือซึ่งกันและกัน

ใช้เวลาโฮ่วของ กิจกรรมการแสดงผลงาน แอปเป็นวิธีที่ยอดเยียมในการโปรโมทชมรม ไอเดียแอป และทักษะการเขียนโค้ดในหมู่เพื่อน ครอบครัว นักการศึกษา และชุมชน และยังอาจช่วยให้คุณหาสมาชิกเพิ่มได้ด้วย ดูเคล็ดลับการจัดแสดงแอปของคุณได้ที่หน้า 12



แชร์ไอเดีย สมาชิกบางคนอาจสนใจอยากพัฒนาเกม ในขณะที่คนอื่นอาจต้องการสร้างแอปเพื่อช่วยเหลือผู้คน เรียนรู้เกี่ยวกับ Swift หรือแม้แต่ควบคุมหุ่นยนต์ ลองคิดหาวิธีให้สมาชิกได้ทำงานร่วมกันในโปรเจกต์ที่ต่างคนต่างก็สนใจดูซิ

จับคู่ให้เหมาะสม บางครั้งสมาชิกที่เก่งกว่าอาจทิ้งคนอื่นไปไกล ลองดูว่ามีสมาชิกคนไหนบ้างที่สามารถจับคู่กับมือใหม่เพื่อช่วยกันเขียนโปรแกรมได้ เพราะการสอนคนอื่นก็เป็นวิธีการเรียนรู้ที่ยอดเยียมไม่แพ้กัน

เรียนรู้และนำไปใช้

1. เรียนรู้เกี่ยวกับ Swift

Swift คือภาษาการเขียนโปรแกรมที่ทรงพลังและใช้งานง่ายที่ Apple พัฒนาขึ้นสำหรับการสร้างแอป ซึ่งเป็นภาษาเดียวกับที่นักพัฒนามืออาชีพในระบบเศรษฐกิจแอปที่เจริญเติบโตอย่างรวดเร็วใช้เพื่อสร้างแอปสำหรับ iPad, Mac, Apple TV, Apple Watch และอื่นๆ Swift ทำให้การเขียนโปรแกรมง่ายขึ้น ยืดหยุ่นมากขึ้น และสนุกยิ่งขึ้น

เรียนรู้เพิ่มเติมเกี่ยวกับ Swift ได้ที่ swift.org

2. เรียนรู้ Xcode และ "พัฒนาใน Swift: การสำรวจ"

เครื่องมือของชมรมถูกสร้างขึ้นเพื่อรองรับโปรเจกต์แอปสำหรับ Xcode ซึ่งเป็นสภาพแวดล้อมการพัฒนาแบบบูรณาการที่นักพัฒนามืออาชีพใช้สร้างแอปจริง ซึ่งได้แก่ โปรแกรมแก้ไขซอร์สโค้ดสำหรับเขียนและจัดการโค้ด โปรแกรมดีบักสำหรับวิเคราะห์ปัญหา และโปรแกรมแก้ไขอินเทอร์เฟซผู้ใช้ที่เรียกว่า Interface Builder สำหรับจัดวางองค์ประกอบภาพของแอปและเชื่อมโยงเข้ากับโค้ด

เรียนรู้เพิ่มเติมเกี่ยวกับ Xcode ได้ที่หน้า [การสนับสนุนเกี่ยวกับ Xcode](#)

"พัฒนาใน Swift: การสำรวจ" จะสอนผู้เรียนผ่านกิจกรรม Playground ใน Xcode ขณะเรียนรู้เรื่องพื้นฐานเกี่ยวกับการโค้ด Playground ใน Xcode ให้คุณเขียนโค้ด Swift และดูผลลัพธ์ได้ทันทีในตัวอย่างแบบสด การเล่นกับโค้ดและดูว่าโค้ดทำอะไรได้บ้างจะช่วยให้คุณเริ่มต้นเขียนโค้ดและทดลองกับไอเดียใหม่ๆ ได้อย่างยอดเยี่ยม

คู่มือผู้สอนสำหรับ "พัฒนาใน Swift: การสำรวจ" มีกิจกรรมเพิ่มเติมที่คุณสามารถใช้กับสมาชิกชมรม เพื่อจุดประกายความสนใจของพวกเขา สนับสนุนการสร้างความเข้าใจ รวมถึงกระตุ้นให้พวกเขากล้าทำและพัฒนาสิ่งที่ตัวเองรัก

ดาวน์โหลดคู่มือผู้สอนสำหรับ "พัฒนาใน Swift: การสำรวจ" >

เรียนรู้และนำไปใช้



[ดาวน์โหลดแอป Xcode >](#)



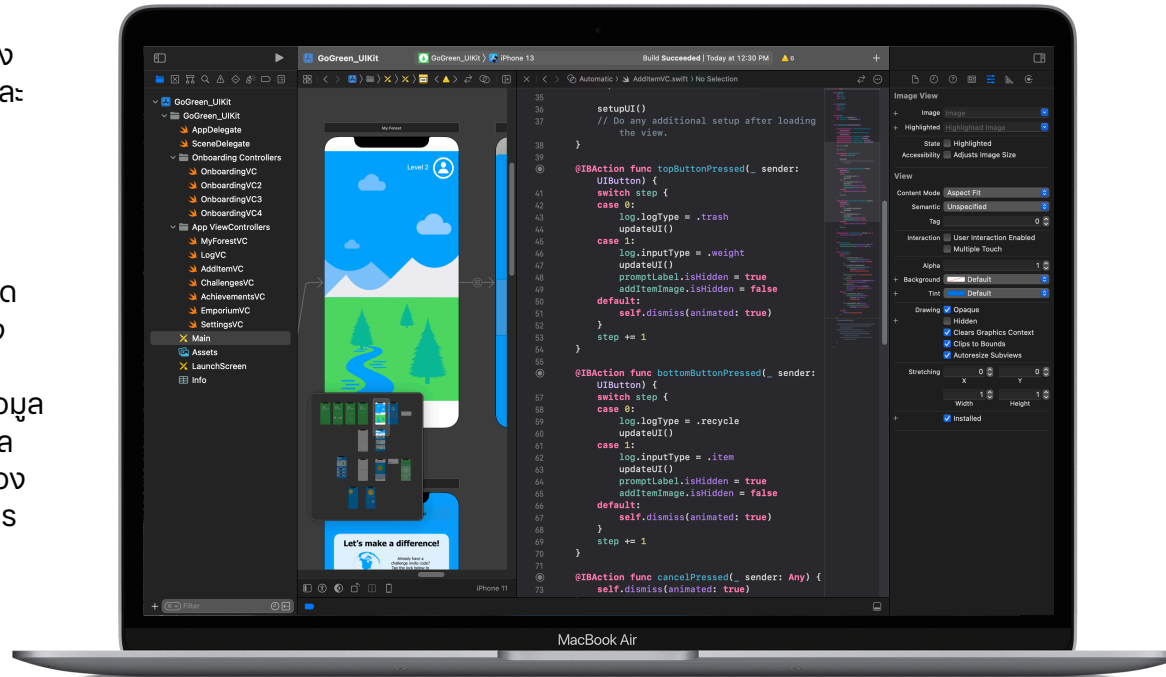
[ดาวน์โหลดคู่มือผู้สอน >](#)

เคล็ดลับสำหรับการเรียนรู้ด้วย Xcode



การเขียนโค้ดไม่ได้มีวิธีเดียว สมาชิกควรตรวจสอบโค้ดของตัวเองและกัน แสดงความคิดเห็น และช่วยติบักให้กันและกัน

ใช้เครื่องมือติบัก เมื่อแอปหยุดทำงาน ไฮไลต์สีแดงจะแสดงบรรทัดโค้ดที่มีปัญหา ใช้ `print()` ในโค้ดเพื่อบันทึกข้อมูลที่มีประโยชน์ลงในจอกอนโซล กำหนดจุดหยุดเพื่อหยุดแอปของคุณชั่วคราว ตรวจสอบตัวแปรและดูโค้ดที่ละบรรทัด



สำรวจการตั้งค่า Xcode ตั้งค่าการแก้ไขข้อความและการตั้งค่าอื่นๆ โดยการเลือก Xcode > Preferences จากแถบเมนู คุณสามารถเพิ่มบัญชีนักพัฒนา ปรับแต่งการนำทาง หรือแบบอักษร เลือกพฤติกรรมบางอย่างเมื่อเกิดเหตุการณ์ และอื่นๆ

เตรียมฝ่ายช่วยเหลือ จัดตั้งพื้นที่ ที่มีผู้เชี่ยวชาญประจำชมรมคอยให้ความช่วยเหลือสมาชิกคนอื่นๆ

หยุดเพื่อคิด บักเป็นสิ่งที่หลีกเลี่ยงไม่ได้ หยุดเพื่อคิดเกี่ยวกับปัญหา มีอาการอะไรบ้าง ทุกอย่างทำงานได้ดีก่อนเกิด [x] หรือไม่

ต่อยอดการเรียนรู้ สมาชิกชมรมชั้นสูงสามารถเรียนรู้หลักสูตร "พัฒนาใน Swift: การสำรวจ" และเน้นไปที่แง่มุมด้านการเขียนโค้ดของคุณแบบแอป Xcode ของตนเองมากยิ่งขึ้น

ปุ่มลัดคีย์บอร์ดหลัก
สร้างและรันโปรเจกต์: `⌘R`
แสดงความคิดเห็นหรือยกเลิก
การแสดงความคิดเห็นบนโค้ดที่เลือก: `⌘/`
ทำให้โค้ดที่เลือกเยื้องอีกครั้ง: `⌘|`
แสดงตัวตรวจสอบ: `⌘\0`
แสดงเอกสารประกอบ: `⌘↑0`



3. เลือกโปรเจกต์ของคุณ

แหล่งข้อมูล Swift Coding Club จะแบ่งเป็นโมดูลโปรเจกต์แอปต่างๆ และความท้าทายในการออกแบบแอป สมาชิกชมรมเรียนรู้แนวทางการเขียนโปรแกรมพร้อมศึกษา Xcode โดยการทำซีรีส์กิจกรรม Playground และโปรเจกต์การพัฒนาแอปที่มีคำแนะนำคอร์ส "พัฒนาใน Swift: การสำรวจ" มีทุกสิ่งที่พวกเขาต้องการในการศึกษาแต่ละโมดูลให้สำเร็จ

สองสามโปรเจกต์แอปแรกไม่ต้องใช้ความรู้ที่มีก่อนหน้า แล้วจากนั้นความยากจะเพิ่มขึ้นในโปรเจกต์ต่อๆ ไป ตรวจสอบความซับซ้อนของแต่ละโปรเจกต์และเลือกจุดเริ่มต้นที่เหมาะสมกับประสบการณ์การเขียนโค้ดของสมาชิกชมรม

ความท้าทายในการออกแบบแอปได้รับการออกแบบมาให้ทำคู่กับโมดูลอื่นหรือทำแยกต่างหาก

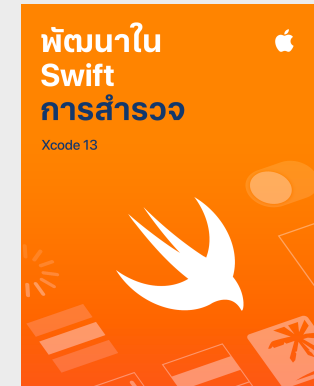
โมดูลที่ 1: แอป PhotoFrame

โมดูลที่ 2: แอป QuestionBot

โมดูลที่ 3: แอป ColorMix

โมดูลที่ 4: แอป ElementQuiz

โมดูลที่ 5: ความท้าทายในการออกแบบแอป



[ดาวโหลดคอร์ส >](#)



4. ความท้าทายในการออกแบบแอป

ขณะที่สมาชิกชมรมกำลังเรียนรู้การสร้างแอปใน Xcode พวกเขาก็ยังสามารถเริ่มวางแนวคิดและแชร์ไอเดียเกี่ยวกับแอปที่ต้องการจะสร้าง และช่วยเปลี่ยนไอเดียให้กลายเป็นแอปต้นแบบ ทดสอบแอปกับเพื่อนๆ และปรับปรุงประสบการณ์ของผู้ใช้งาน ความท้าทายในการออกแบบแอปจะช่วยฝึกฝนมากกว่าทักษะการเขียนโปรแกรม แต่ยังทำให้สมาชิกได้ปลดปล่อยความคิดสร้างสรรค์และความช่างประดิษฐ์ของตนเอง

สมาชิกชมรมจะได้เรียนรู้เกี่ยวกับคุณสมบัติของแอปที่ประสบความสำเร็จ และข้อที่ควรคำนึงเมื่อออกแบบแอปของตัวเอง พวกเขาจะได้ทำงานในสมุดงานออกแบบแอป "พัฒนาใน Swift" ซึ่งเป็นส่วนหนึ่งในกระบวนการออกแบบแอปและพัฒนาต้นแบบแอปที่ใช้งานได้ ที่ตนเองสามารถนำแชร์ในการแสดงผลงานแอป สมาชิกชมรมสามารถทำโปรเจกต์แอปในแต่ละสัปดาห์สักระยะหนึ่ง แล้วสร้างไอเดียแอปของตัวเองในเวลาที่เหลือของสัปดาห์ หรือสลับสัปดาห์ระหว่างโปรเจกต์





5. ต่อยอดการเรียนรู้

คุณยังสามารถเพิ่มเซสชันตามความสนใจของสมาชิกได้อีกด้วย คุณสามารถต่อยอดกิจกรรมการออกแบบและการเขียนโค้ดโดยการสำรวจแอปสำหรับอุปกรณ์ที่เชื่อมต่อกันหรือสำหรับ watchOS หรือพูดถึงหัวข้อที่ลึกขึ้นอย่างการเรียนรู้ของระบบและเทคโนโลยีความจริงเสริมเพื่อสำรวจอนาคตการออกแบบแอป

หากต้องการกระตุ้นให้เกิดการระดมสมองเกี่ยวกับการออกแบบมากขึ้น คุณอาจเชิญวิทยากรหรือเพิ่มกิจกรรมทัศนศึกษา เพื่อช่วยให้สมาชิกชมรมเข้าใจกลุ่มเป้าหมายและข้อกำหนดในการออกแบบสำหรับโปรเจกต์ได้ดียิ่งขึ้น



เฉลิมฉลอง



กิจกรรมในชุมชนหรือการแสดงผลงานแอปจากระยะไกล

จัดกิจกรรมในชุมชนหรือการแสดงผลงานแอปจากระยะไกล เพื่อเปิดโอกาสให้ผู้คนในชุมชนมีส่วนร่วมในวงกว้างขึ้น และศึกษาความเป็นไปได้ที่โค้ดจะช่วยแก้ปัญหาที่เผชิญอยู่ในปัจจุบัน กิจกรรมเหล่านี้ยังเป็นวิธีอันยอดเยี่ยมในการแสดงความสามารถของสมาชิกชมรมอีกด้วย

1. วางแผนงานจัดแสดงครั้งใหญ่ กำหนดวันที่และเชิญเพื่อน นักการศึกษา ครอบครัว และสมาชิกชมรม

จัดสรรเวลาให้แต่ละทีมนำเสนอโปรเจกต์ของตน และจัดเซสชันถามตอบสั้นๆ แบบตัวต่อตัวหรือแบบออนไลน์ ถ้ากลุ่มมีขนาดใหญ่ ก็แบ่งออกเป็น 2 รอบให้สมาชิกดูการนำเสนอของกันและกันได้

แนะนำให้ปิดท้ายด้วยสไลด์โชว์รูปถ่ายสนุกๆ จากแต่ละเซสชันของชมรม

2. รางวัลการออกแบบ การแข่งขันฉันทน์มิตรคือเครื่องสร้างแรงจูงใจชั้นเยี่ยม สร้างแรงบันดาลใจให้สมาชิกชมรมด้วยการมอบรางวัลเพื่อยกย่องจุดเด่นด้านใดด้านหนึ่งในการเขียนโค้ดและการออกแบบ ตัวอย่างเช่น

- วิศวกรรมดีเด่น
- นวัตกรรมดีเด่น
- การออกแบบดีเด่น
- การนำเสนอดีเด่น

นอกจากนี้ คุณยังสามารถกระตุ้นการมีส่วนร่วมของผู้ชมด้วยรางวัลขวัญใจมหาชนได้อีกด้วย



ดาวน์โหลดและแก้ไข [ประกาศนียบัตร](#) นี้ เพื่อมอบเป็นรางวัลสำหรับด้านต่างๆ



3. คัดเลือกผู้ตัดสินและที่ปรึกษา พวกเขาอาจเป็นนักการศึกษา หรือพนักงาน เพื่อนร่วมงานที่มีความเชี่ยวชาญในการเขียนโค้ด ผู้เชี่ยวชาญจากวงการนักพัฒนาหรือการออกแบบ ผู้นำชุมชน ท้องถิ่น หรือบุคคลที่จะได้ประโยชน์จากไอเดียโปรเจกต์

ผู้ตัดสินสามารถพบปะพูดคุยกับสมาชิกชมรมได้เลยโดยไม่จำเป็นต้องรอนถึงวันจัดแสดงผลงาน หรือคุณจะใช้ผู้ตัดสินมาเป็นวิทยากรรับเชิญเพื่อให้ความรู้ ในขณะที่ผู้เรียนอยู่ในขั้นตอนการระดมสมองหรือวางแผนโปรเจกต์ก็ได้

4. แบ่งปันและสร้างแรงบันดาลใจ คุณอาจบันทึกวิดีโองานนำเสนอเก็บไว้ก็ได้ แล้วนำไปเผยแพร่ต่อให้ให้คนอื่นๆ ในชุมชนดู พร้อมกับจัดทำวิดีโอไฮไลท์เพื่อเป็นแรงบันดาลใจให้กับสมาชิกชมรมรุ่นต่อๆ ไป





พัฒนาใน Swift
Swift Coding Club

ประกาศนียบัตรรับรองความสำเร็จ

มอบให้กับ

สำหรับ

ลายเซ็น

วันที่

โมดูล Swift Coding Club

โมดูลที่ 1: แอป PhotoFrame

โมดูลที่ 2: แอป QuestionBot

โมดูลที่ 3: แอป ColorMix

โมดูลที่ 4: แอป ElementQuiz

โมดูลที่ 5: ความท้าทายในการออกแบบแอป



ແອປ PhotoFrame

ໂມດູລ໌ 1



แอป PhotoFrame

โมดูลที่ 1 ภาพรวม

ค้นหว่าการสร้างแอปแรกนั้นเป็นเรื่องง่ายแค่ไหน ในโมดูลนี้ คุณจะได้เรียนรู้แนวคิดหลักและทักษะที่ต้องใช้ในการสร้างแอปที่สามารถแสดงองค์ประกอบ UI พื้นฐาน เช่น รูปภาพ การเข้าใจพื้นฐานขององค์ประกอบ UI เป็นเรื่องสำคัญในการสร้างแอป ซึ่งจะเป็นประโยชน์เมื่อคุณต่อยอดทักษะการเขียนโค้ดและการพัฒนาแอปเพิ่มขึ้น ในการทำโปรเจกต์นี้ คุณยังจะได้ทำความคุ้นเคยกับ Xcode, Interface Builder และโปรแกรมจำลอง อีกทั้งยังได้เรียนรู้วิธีใช้ทั้งหมดนี้ร่วมกันเพื่อสร้างแอปของคุณ

● เซสชัน 1-7

เรียนรู้เกี่ยวกับค่าและทดลองใช้ค่า ตัวแปร และค่าคงที่ใน Playground ของ Xcode

- พื้นฐานของ Playground
- การตั้งชื่อและตัวระบุ
- สตริง
- ค่าคงที่และตัวแปร

● เซสชัน 8-9

ใช้ทักษะและแนวคิดใหม่ๆ เพื่อสร้าง Playground สำหรับเกมคำศัพท์

● เซสชัน 10-12

สร้างแอป PhotoFrame ใน Xcode และ Interface Builder



แอป PhotoFrame

1 เริ่มต้นด้วย Xcode

ศึกษาแนวทางต่างๆ เกี่ยวกับ Playground ใน Xcode และเรียนรู้วิธีเข้าไปเปลี่ยนแปลงโค้ดในโปรแกรมพื้นฐาน

เริ่มเลย: ศึกษาพื้นฐานเกี่ยวกับการเขียนโปรแกรมและบทบาทของอินพุตและเอาต์พุตข้อมูล

การเขียนโปรแกรมคืออะไร (หน้า 15)
คำ (หน้า 16–17)

เล่น: เรียนรู้วิธีเข้าไปเปลี่ยนแปลงโค้ด Playground เกี่ยวกับเรื่องพื้นฐานของ Playground (หน้า 27–29)

2–3 การตั้งชื่อและตัวระบุ

เรียนรู้เกี่ยวกับความสำคัญของการตั้งชื่อในการเขียนโปรแกรม และสร้างโปรแกรมง่ายๆ เพื่อแก้ปัญหา

เริ่มเลย: ค้นหาว่าทำไมการตั้งชื่อและตัวระบุจึงสำคัญ และตั้งชื่อองค์ประกอบหลักสำหรับเกมใหม่ขณะร่างแบบเพื่อสร้างเกม

การตั้งชื่อและตัวระบุ (หน้า 18–19)

เล่น: ฝึกฝนทักษะการตั้งชื่อขณะสร้างโปรแกรมเพื่อแก้ปัญหาต่างๆ
Playground เกี่ยวกับการตั้งชื่อและตัวระบุ (หน้า 30–33)

4–5 สตริง

เรียนรู้เกี่ยวกับสตริงและวิธีการใช้สตริงในโค้ดเพื่อสร้างเกมง่ายๆ

เริ่มเลย: ระบุคุณสมบัติหลักของสตริงและสร้างการตอบกลับของแชทบอทของคุณเอง
สตริง (หน้า 20–21)

เล่น: สร้างเกมเติมคำในช่องว่าง
Playground เกี่ยวกับสตริง (หน้า 38–40)

แอป PhotoFrame

6-7 คำคงที่และตัวแปร

เรียนรู้วิธีประกาศตัวแปรและคำคงที่ และสร้างโปรแกรมเพื่อติดตามคะแนน

เริ่มเลย: เปรียบเทียบตัวแปรและคำคงที่ และลองจินตนาการว่าชีวิตของคุณเป็นโปรแกรม คำคงที่และตัวแปร (หน้า 22)

เล่น: สร้างโปรแกรมเพื่อช่วยติดตามคะแนนเกม Playground เกี่ยวกับคำคงที่และตัวแปร (หน้า 41-44)

8-9 เกมคำศัพท์

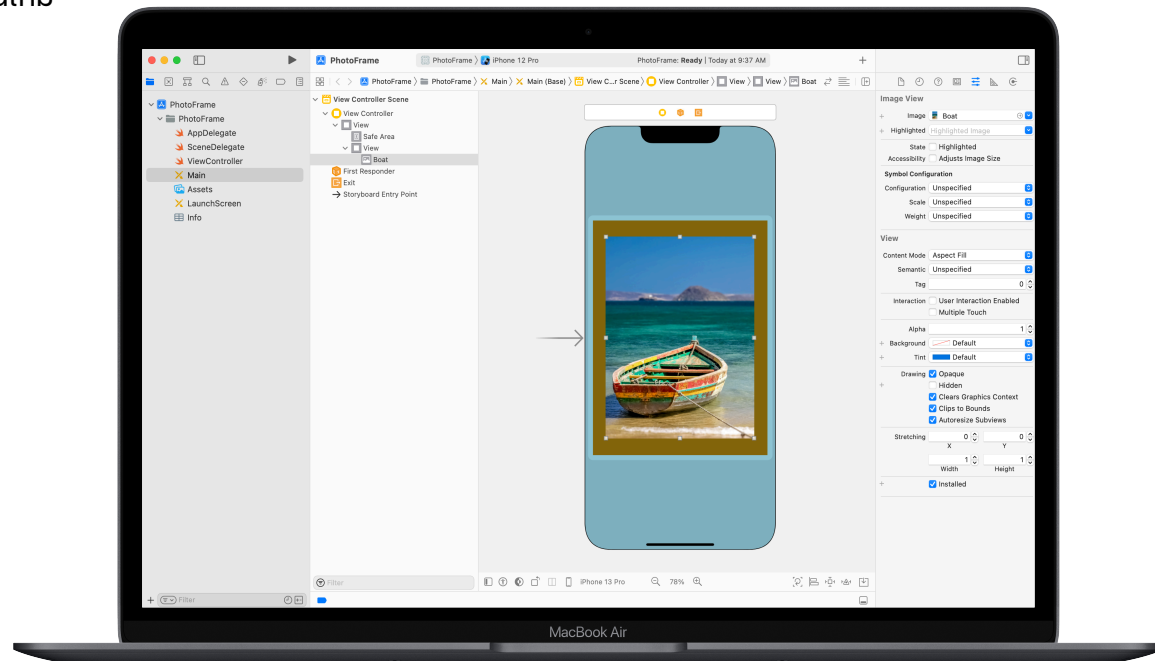
ใช้ความรู้เกี่ยวกับคำ คำคงที่ และสตริงเพื่อสร้างเกมคำศัพท์ใน Playground จากนั้นให้สมาชิกชมรมลองทำ

นำไปใช้: สร้างเกมแทนที่คำศัพท์ที่นำไปสู่เรื่องราวที่ตลกขบขัน Playground เกี่ยวกับเกมคำศัพท์ (หน้า 45-46)

10-12 แอป PhotoFrame

เรียนรู้วิธีใช้ Interface Builder ใน Xcode เพื่อสร้างและใช้งานแอปง่ายๆ

นำไปใช้: สร้างและดูแอปที่แสดงรูปภาพในกรอบที่กำหนดเอง โปรเจกต์แอป PhotoFrame (หน้า 48-73)



แอป QuestionBot

โมดูลที่ 2



แอป QuestionBot

ภาพรวมของโมดูลที่ 2

คุณเคยใช้แอปแบบทดสอบหรือเคยสงสัยหรือไม่ว่า Siri ทำงานอย่างไร ทุกแอปล้วนมีตรรกะภายในที่กำหนดลักษณะการทำงาน QuestionBot ให้คุณสร้างแอปที่มี "สมอง" ของบอทที่ตอบกลับคำถามต่างๆ ได้หลากหลายรูปแบบ เพื่อที่จะทำสิ่งนี้ คุณจะต้องเรียนรู้วิธีออกแบบอัลกอริทึม จัดกลุ่มโค้ดในฟังก์ชัน ใช้ประเภทที่แตกต่างกัน และอื่นๆ โมดูลนี้จะช่วยเน้นให้คุณเข้าใจวิธีการทำงานของแอป ได้เรียนรู้เกี่ยวกับการสร้างโค้ดที่ควบคุมอินเทอร์เฟซของแอป และสำรวจตรรกะที่คุณสามารถใช้เพื่อสร้างแอปที่จำลองความฉลาดของมนุษย์

● เซสชัน 1-10

เรียนรู้เกี่ยวกับอัลกอริทึมที่เป็นหัวใจหลักของการเขียนโปรแกรม และทดลองกับฟังก์ชัน ประเภท และพารามิเตอร์ใน Playground ใน Xcode

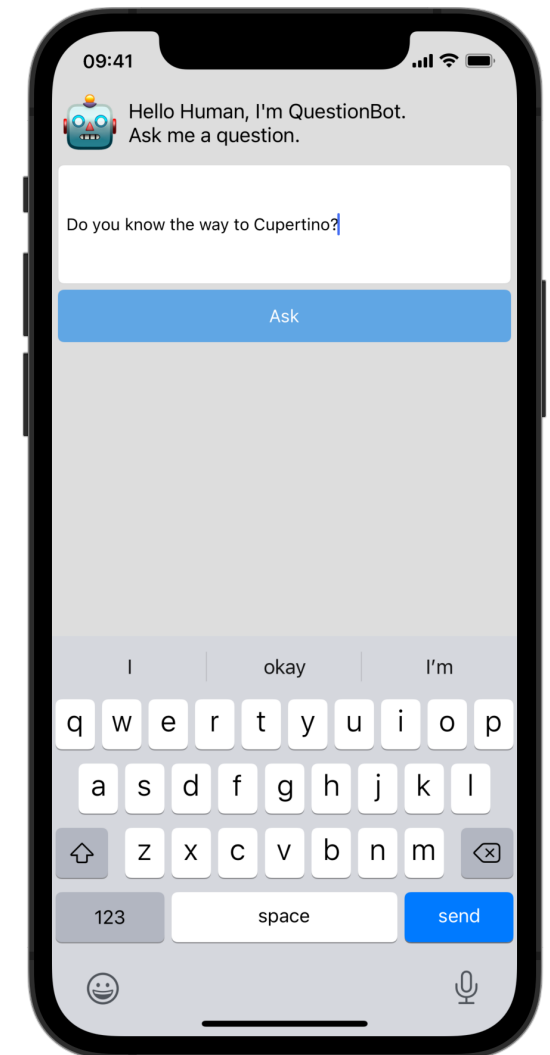
- อัลกอริทึม
- ฟังก์ชัน
- ประเภท
- พารามิเตอร์
- การตัดสินใจด้วยบูลีน

● เซสชัน 11-12

นำทักษะและแนวคิดใหม่ไปใช้สร้างรูปแบบการเดินใน Playground เกี่ยวกับ BoogieBot

● เซสชัน 13-14

ใน Xcode ให้เพิ่มฟังก์ชันเพื่อตั้งโปรแกรมให้ "สมอง" ของแอป QuestionBot สามารถตอบคำถามได้



แอป QuestionBot

1-2 อัลกอริทึม

เรียนรู้เกี่ยวกับอัลกอริทึมในฐานะที่เป็นเครื่องมือหลักในการเขียนโปรแกรม และฝึกออกแบบอัลกอริทึมเพื่อแก้ปัญหาต่างๆ ในชีวิตประจำวัน

เริ่มเลย: ใช้การจัดลำดับและการเลือกในอัลกอริทึมเพื่อแก้ปัญหาต่างๆ และออกแบบอัลกอริทึมสำหรับโปรแกรมที่จับคู่เพลงให้เข้ากับอารมณ์ของคุณ

อัลกอริทึม (หน้า 109)

การจัดลำดับ (หน้า 110)

การเลือก (หน้า 111)

3-4 ฟังก์ชัน

เรียนรู้เกี่ยวกับฟังก์ชันการสร้างส่วนโค้ดที่ใช้ซ้ำได้ และสร้าง โปรแกรมสร้างเสียงเพลง

เริ่มเลย: ฝึกฝนการจัดกลุ่มคำแนะนำภายใต้ฟังก์ชันในบริบทที่คุณเคย เช่น การเตรียมอาหารค่ำ ฟังก์ชัน (หน้า 112-114)

เล่น: สร้างโปรแกรมที่สร้างเสียงเพลงซ้ำๆ Playground เกี่ยวกับฟังก์ชัน (หน้า 121-124)

5-6 ประเภท

เรียนรู้เกี่ยวกับประเภทเพื่อช่วยแยกแยะข้อมูลประเภทต่างๆ และสร้างโปรแกรมเพื่อทำการคำนวณง่ายๆ

เริ่มเลย: ศึกษาประเภทเพื่ออธิบายค่า และพิจารณาประเภทของส่วนประกอบที่คุณจะใช้ในงานการสร้าง

ประเภท (หน้า 115-116)

เล่น: สร้างโปรแกรมเพื่อทำการคำนวณง่ายๆ ให้สำเร็จ

Playground เกี่ยวกับประเภท (หน้า 125-127)

แอป QuestionBot

7-8 พารามิเตอร์

เรียนรู้เกี่ยวกับการใช้พารามิเตอร์เพื่อกำหนดอินพุตสำหรับฟังก์ชัน และสร้างโปรแกรมที่ผลิตประโยคที่แตกต่างกันตามค่าที่คุณป้อน

เริ่มเลย: ใช้พารามิเตอร์เพื่อทำให้ฟังก์ชันยืดหยุ่นมากขึ้น และปรับปรุงฟังก์ชันการเตรียมอาหารคำให้เป็นไปตามความต้องการที่เฉพาะเจาะจงพารามิเตอร์ (หน้า 116-117)

เล่น: สร้างโปรแกรมที่ใช้ฟังก์ชันเพื่อผลิตประโยคที่แตกต่างกันตามค่าที่คุณใส่เข้าไป Playground เกี่ยวกับพารามิเตอร์และผลลัพธ์ (หน้า 128-130)

9-10 การตัดสินใจ ด้วยบูลีน

เรียนรู้ว่าประเภทบูลีนนั้นยอดเยี่ยมแค่ไหนในการเขียนโปรแกรม และสร้างโปรแกรมเพื่อตัดสินใจว่าปีที่กำหนดเป็นปีอธิกสุรทินหรือไม่

เริ่มเลย: สำรวจและใช้บูลีนเพื่อช่วยหุ่นยนต์แก้ไขสถานการณ์ที่ซับซ้อน การตัดสินใจด้วยบูลีน (หน้า 118)

เล่น: สร้างโปรแกรมที่ตัดสินใจว่าปีดังกล่าวเป็นปีอธิกสุรทินหรือไม่ Playground เกี่ยวกับการตัดสินใจ (หน้า 131-134)

11-12 BoogieBot

ใช้ฟังก์ชันเพื่อสร้างรูปแบบการเต้นที่ซับซ้อนจากส่วนย่อยๆ และแชร์ท่าเต้นของคุณกับคนอื่นๆ เป็นภาพเคลื่อนไหว

นำไปใช้: สร้างรูปแบบการเต้นสำหรับ BoogieBot และบันทึกผลงานสร้างสรรค์ของคุณเป็นภาพเคลื่อนไหว Playground เกี่ยวกับ BoogieBot (หน้า 135)

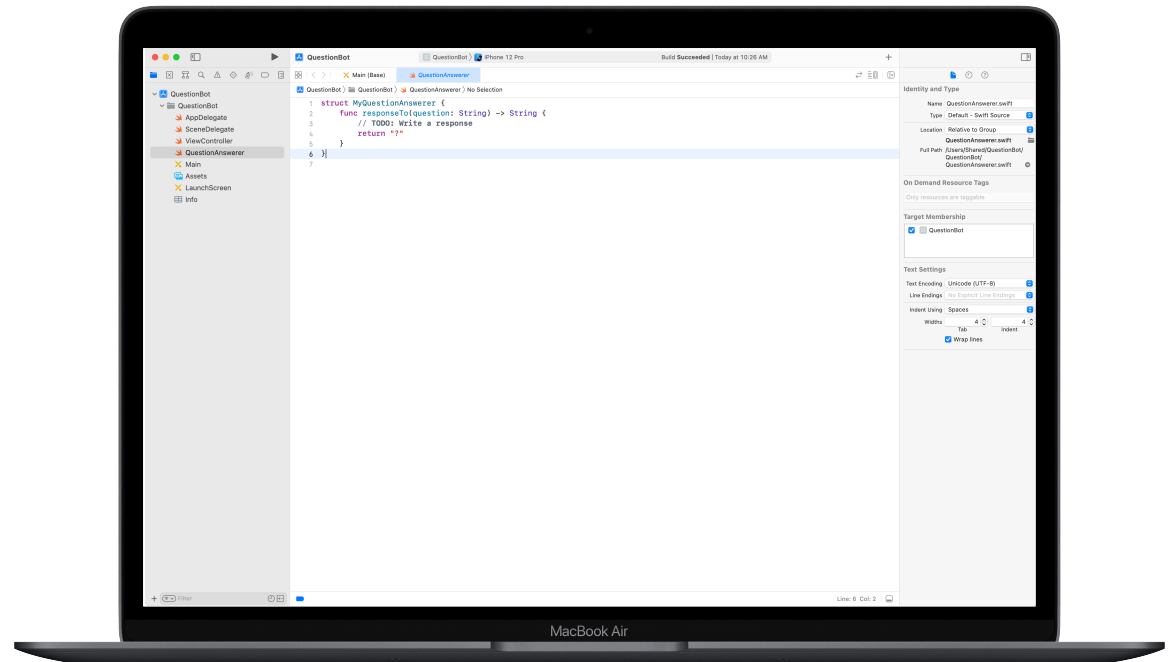
แอป QuestionBot

13-14 QuestionBot

สร้างตรรกะของแอป QuestionBot เพื่อให้ตอบคำถามต่างๆ ได้หลากหลายรูปแบบ

นำไปใช้: ตั้งโปรแกรม "สมอง" ของแอป QuestionBot ให้ตัดสินใจว่าจะตอบคำถามอย่างไร และเรียนรู้วิธีทดสอบและแก้ปัญหาโค้ดของคุณ

โปรเจกต์แอป QuestionBot (หน้า 138-150)



แอป ColorMix

โมดูลที่ 3



แอป ColorMix

ภาพรวมของโมดูลที่ 3

มาถึงอินเทอร์เฟซผู้ใช้ (UI) ของ iPhone จนถึงตอนนี้ คุณได้สร้างแอปจากองค์ประกอบ UI พื้นฐาน และได้เรียนรู้วิธีสร้างตรรกะเบื้องหลัง UI โดย ColorMix จะช่วยให้คุณเรียนรู้วิธีสร้างแอปแบบอินเทอร์แอคทีฟที่มีส่วนควบคุมอย่างปุ่มและสวิตช์ ที่สำคัญกว่านั้นก็คือ คุณจะได้เรียนรู้วิธีเชื่อมโยงองค์ประกอบ UI เสมือนเหล่านั้นกับโค้ด Swift เพื่อให้ทำงานตามต้องการ เพื่อที่จะทำสิ่งนี้ คุณจะต้องเรียนรู้เกี่ยวกับการกำหนดประเภทแบบกำหนดเองที่มีพรีอพเพอร์ทีและเมธอดของคุณเอง โดยใช้อินสแตนซ์ของประเภทและรวบรวมข้อมูลในอาร์เรย์ ท้ายที่สุด คุณจะมี ColorMix ซึ่งเป็นแอปที่สร้างถึงสี่อื่นๆ โดยการผสมสีแดง เขียว และน้ำเงิน

เซสชัน 1-6

เรียนรู้เกี่ยวกับการจัดเรียงข้อมูล ทดลองกำหนดเมธอดและพรีอพเพอร์ทีสำหรับประเภทที่กำหนดเอง และใช้งานอาร์เรย์ใน Playground ใน Xcode

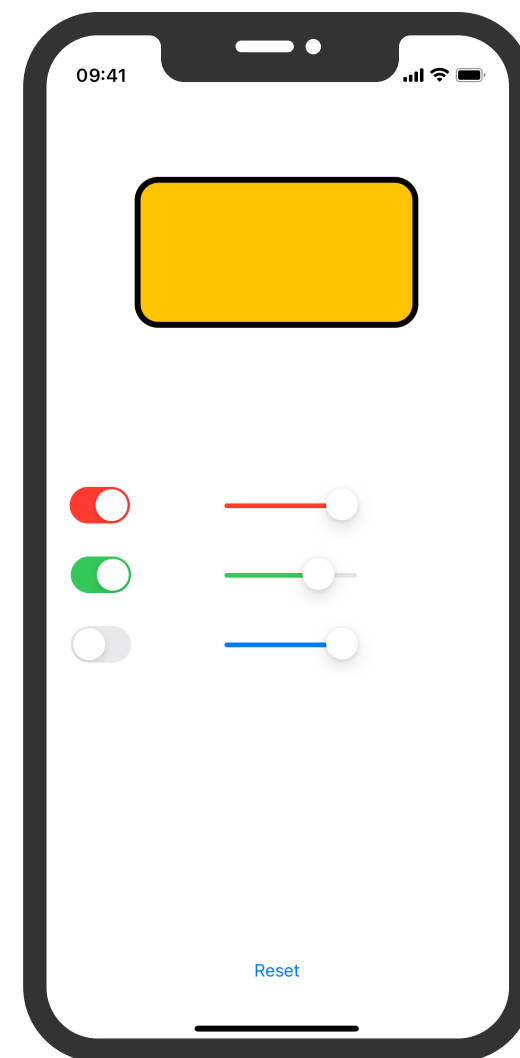
- อินสแตนซ์ เมธอด และพรีอพเพอร์ที
- อาร์เรย์และลูป
- สตริงค์

เซสชัน 7-8

เรียนรู้วิธีสร้างกราฟิก จากนั้นสร้างกราฟิก อีโมจิ และการวาดเส้นเคลื่อนไหวแบบทีละพิกเซล

เซสชัน 9-12

สร้างแอป ColorMix โดยการเพิ่มสวิตช์และตัวเลื่อนลงใน UI



แอป ColorMix

1-2

อินสแตนซ์, เมรอด และพรีออฟเพอร์ตี

เรียนรู้การสร้างอินสแตนซ์ของประเภท และวิธีใช้เมรอดและพรีออฟเพอร์ตีต่างๆ แล้วเขียนโปรแกรมการแข่งขันของหุ่นยนต์

เริ่มเลย: ศึกษาว่าประเภทกำหนดเมรอดและพรีออฟเพอร์ตีอย่างไร และทดลองอธิบายเมรอดและพรีออฟเพอร์ตีกับสัตว์ประเภทต่างๆ

อินสแตนซ์, เมรอด และพรีออฟเพอร์ตี (หน้า 184)

เล่น: สร้างโปรแกรมเพื่อตั้งค่าการแข่งขันของหุ่นยนต์สองตัว

Playground เกี่ยวกับอินสแตนซ์, เมรอด และพรีออฟเพอร์ตี (หน้า 196–198)

3-4

อาร์เรย์และลูป

เรียนรู้วิธีจัดเรียงข้อมูลในอาร์เรย์ และวิธีประมวลผลอาร์เรย์โดยใช้ลูป และสร้างโปรแกรมเพื่อนับคะแนนโหวต ติดตามความก้าวหน้า และค้นหาคำสำคัญ

เริ่มเลย: ใช้การทำซ้ำในอัลกอริทึม และใช้ลูปเพื่อทำงานกับอิลิเมนต์ในอาร์เรย์ สร้างอัลกอริทึมเพื่ออธิบายวิธีเล่นเกมกระดาน และพิจารณาวิธีที่คุณสามารถทำงานกับคอลเลกชัน

รายการและอาร์เรย์ (หน้า 185)

อัลกอริทึม: การทำซ้ำ (หน้า 186–187)

ลูป (หน้า 188)

ใช้งานอาร์เรย์—การค้นหา (หน้า 189–190)

เล่น: สร้างโปรแกรมเพื่อนับคะแนนโหวต ติดตามความก้าวหน้าของเป้าหมายรายวัน และกรองข้อความเพื่อหาคำสำคัญ

Playground เกี่ยวกับอาร์เรย์และลูป (หน้า 199–202)

5-6

สตริงค์

เรียนรู้วิธีสร้างประเภทที่กำหนดเองด้วยสตริงค์ และใช้ประเภทที่กำหนดเองเพื่อแก้ปัญหาคณิตศาสตร์

เริ่มเลย: กำหนดประเภทของตัวเองด้วยสตริงค์ และสร้างประเภทที่กำหนดเองสำหรับสัตว์ที่คุณเลือก

กำหนดประเภทของคุณเองด้วยสตริงค์ (หน้า 191–192)

เล่น: สร้างโปรแกรมที่ใช้ประเภทที่กำหนดเองเพื่อแก้ปัญหาคณิตศาสตร์

Playground เกี่ยวกับโครงสร้าง (หน้า 203–205)

แอป ColorMix

7-8 ศิลปะพิกเซล

เรียนรู้วิธีสร้างกราฟิก จากนั้นสร้างกราฟิกของตัวเองแบบทีละพิกเซล

นำไปใช้: เขียนโค้ดเพื่อสร้างสรรค์กราฟิกอีโมจิ และการวาดเส้นเคลื่อนไหว

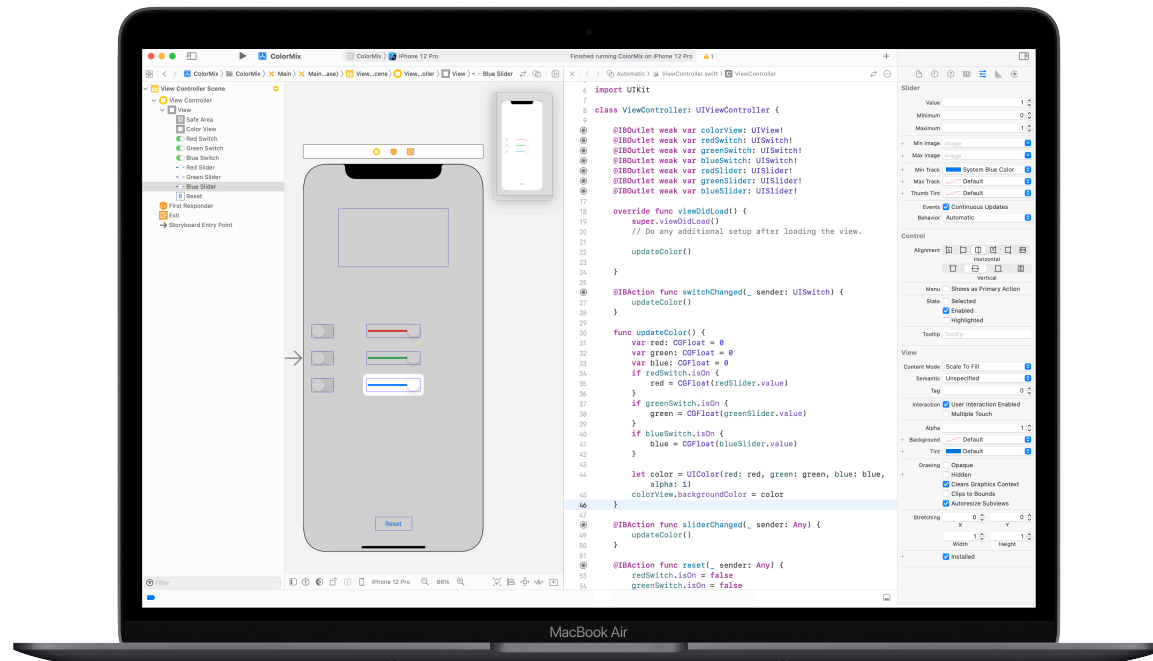
Playground เกี่ยวกับศิลปะพิกเซล (หน้า 215–216)

9-12 ตัวเลือกสี

เรียนรู้วิธีการใช้แอ็คชันและเอทก์เล็ตเพื่อเชื่อมโยงโค้ด Swift กับ UI ของแอป

นำไปใช้: สร้างแอปที่มีสวิตช์ ตัวเลื่อน และปุ่มที่ให้คุณผสมสีของตัวเอง

โปรเจกต์แอป ColorMix (หน้า 301–345)



ແອປ ElementQuiz

ໂມດູນທີ 4



แอป ElementQuiz

ภาพรวมของโมดูลที่ 4

คนส่วนใหญ่ใช้แอปเพื่อแก้ปัญหาบางอย่าง เช่น ช่วยจัดระเบียบชีวิต คำนวณการเงิน หรือถามเส้นทาง ใน ElementQuiz คุณจะสร้างแอปที่ช่วยให้คุณจดจำธาตุในตารางธาตุ คุณจะเรียนรู้เกี่ยวกับ enum จากนั้นนำสิ่งที่คุณเรียนรู้ไปจัดการกับปัญหาแอปแต่ละอย่างตามที่เลือก คุณสามารถสร้างแอป MemeMaker สร้างเกมเป่ายิงฉุบ หรือพัฒนา ElementQuiz ให้ดียิ่งขึ้นไปอีกระดับ

● เซสชัน 1-4

ทำตามขั้นตอนเพื่อสร้างอินเทอร์เฟซเฟลชการ์ดของแอป ElementQuiz

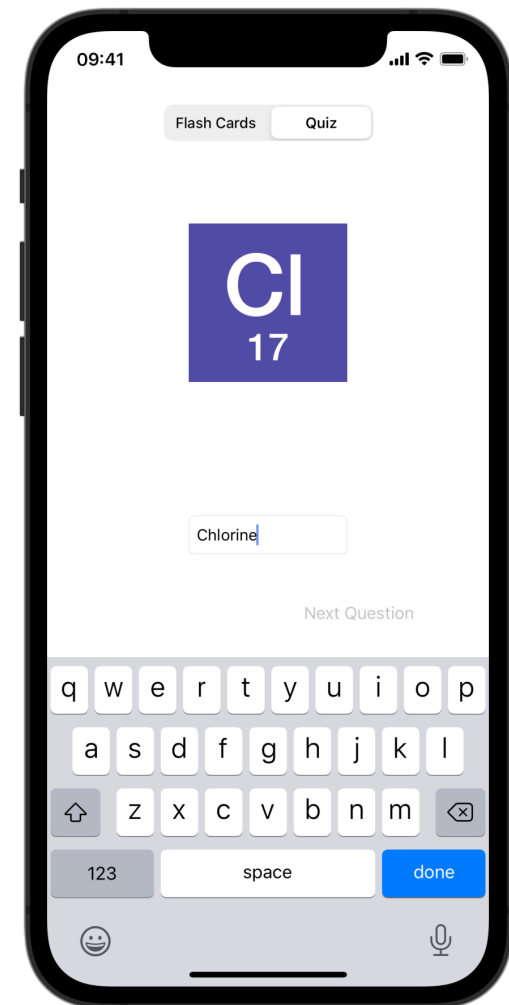
● เซสชัน 5-6

เรียนรู้เกี่ยวกับ enum และสร้างโปรแกรมเพื่อนับคะแนนโหวต

- Enum และ Switch

● เซสชัน 7-12

เลือกหนึ่งในสามโปรเจกต์แอปที่จะสร้าง



แอป ElementQuiz

1-4

แอป ElementQuiz: ส่วน ก

เรียนรู้วิธีสร้างแอปแฟลชการ์ดที่ช่วยคุณจดจำธาตุในตารางธาตุ

นำไปใช้: สร้างแอปแบบทดสอบด้วยอินเทอร์เฟซแฟลชการ์ด ที่ช่วยผู้ใช้งานเรียนรู้สัญลักษณ์ทางเคมีของธาตุต่างๆ

โปรเจกต์แอป ElementQuiz ส่วนที่ 1-3 (หน้า 401-418)

5-6

Enum และ Switch

เรียนรู้เกี่ยวกับ enum และสร้างโปรแกรมนับผลโหวต

เล่น: สร้างโปรแกรมนับผลโหวต

Playground เกี่ยวกับ Enum และ Switch (หน้า 206-208)

แอป ElementQuiz

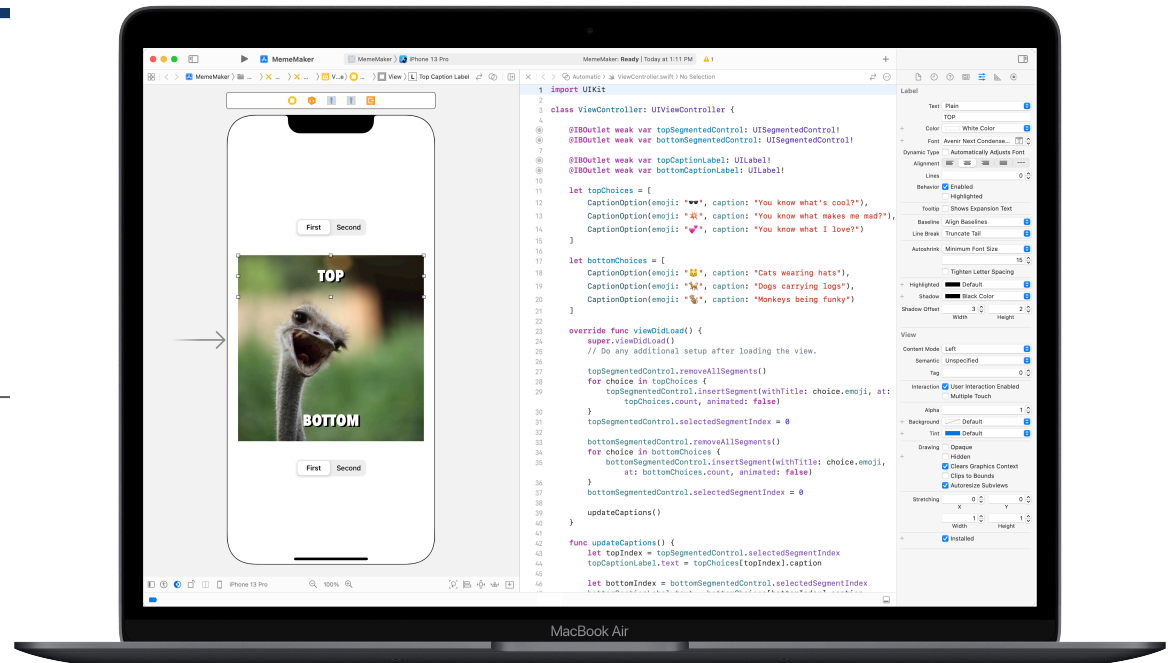
สำหรับเซสชัน 7-12 ให้เลือกหนึ่งในสามโปรเจกต์แอปต่อไปนี้ ดาวบ่งบอกถึงระดับความยาก

7-12 แอป MemeMaker

เรียนรู้วิธีใช้ส่วนควบคุมส่วนต่างๆ เพื่อแสดงคำบรรยายที่แตกต่างกันเหนือและใต้รูป เนื่องจากส่วนควบคุมนั้นแยกกัน คุณจึงสามารถจับคู่และผสมข้อความเพื่อสร้างชุดข้อความแบบกำหนดเองได้ เรียนรู้วิธีใช้ตัวจำกัดทางเพื่อให้ผู้ใช้งานสามารถลากคำบรรยายไปมาทั่วหน้าจอได้

นำไปใช้: สร้างแอปที่ให้คุณเพิ่มคำบรรยายสนุกๆ ที่กำหนดเองได้ลงในรูปถ่ายตามอารมณ์ของคุณ

โปรเจกต์แอป MemeMaker (หน้า 384-399)

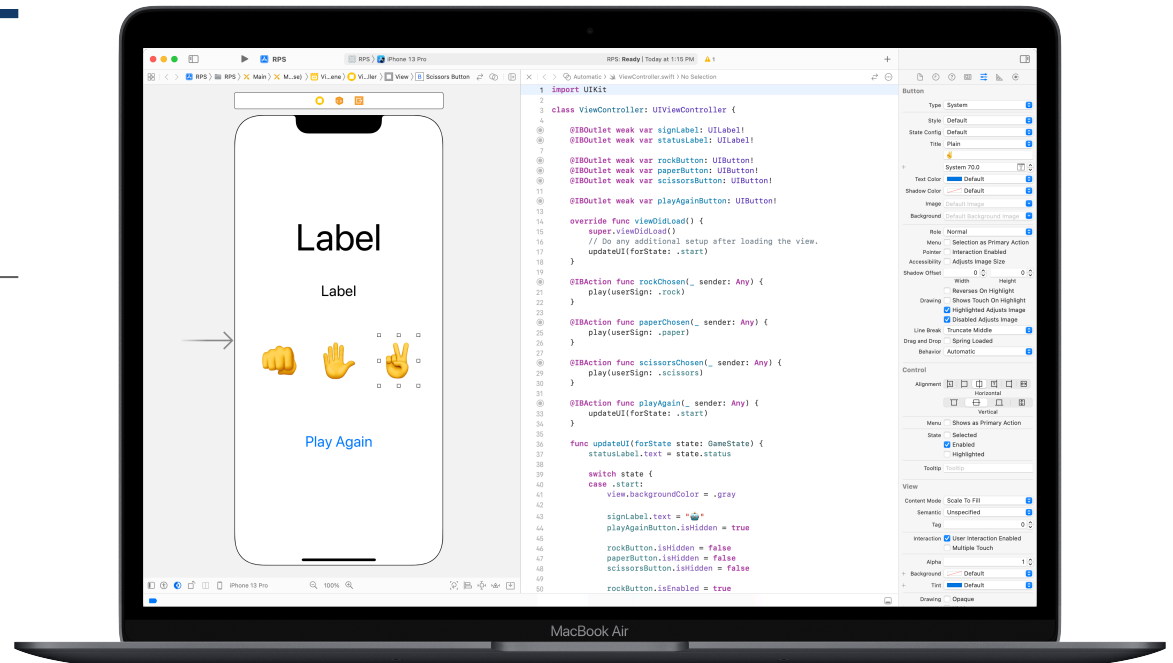


แอป ElementQuiz

7-12 แอปเป่ายิงฉุบ

เรียนรู้วิธีปรับแต่งสตรัคเจอร์และ enum เพื่อสร้างโมเดลและตรรกะสำหรับเกมเป่ายิงฉุบ ใช้ตัวเลขแบบสุ่มเพื่อให้ผู้ใช้งานสามารถแข่งกับคอมพิวเตอร์ได้เรื่อยๆ

นำไปใช้: สร้างแอปเกมโดยใช้อีโมจิและปุ่ม
โปรเจกต์แอปเป่ายิงฉุบ (หน้า 368–383)



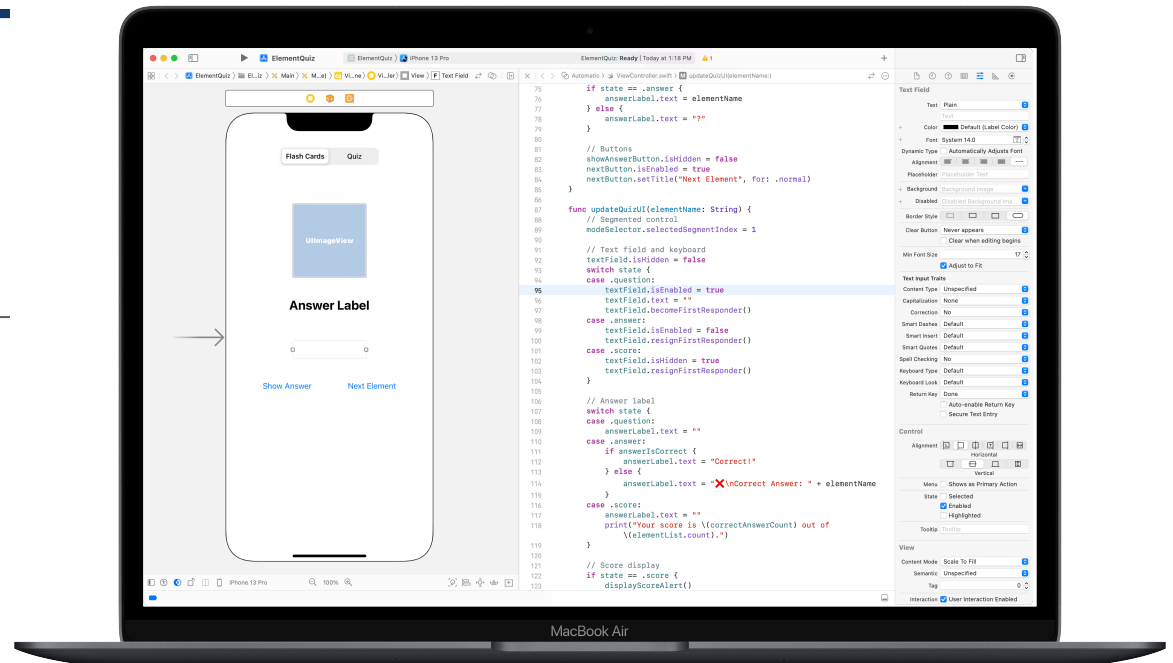
แอป ElementQuiz

7-12 แอป ElementQuiz: ส่วน B

เรียนรู้วิธีการจัดการอินพุตข้อความเพื่อสร้างโหมดแบบทดสอบแบบนับคะแนนในแอป ElementQuiz
เรียนรู้วิธีวางโครงสร้างตรรกะของอินเทอร์เฟซผู้ใช้
และวิธีรีเฟกเตอร์โค้ดของคุณที่เริ่มซับซ้อนขึ้นเรื่อยๆ

นำไปใช้: พัฒนาแอปแบบทดสอบให้มีโหมดแบบทดสอบแบบนับคะแนน

โปรเจกต์แอป ElementQuiz ส่วนที่ 4-10 (หน้า 416-467)



ความท้าทายในการออกแบบแอป

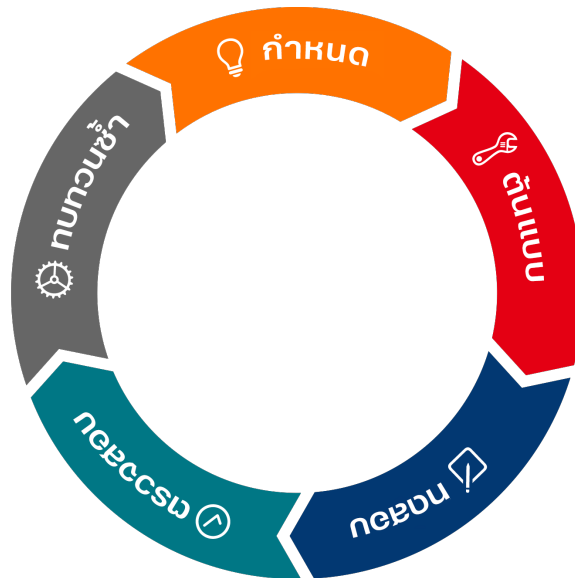
โมดูลที่ 5



ความท้าทายในการออกแบบแอป ภาพรวมของโมดูลที่ 5

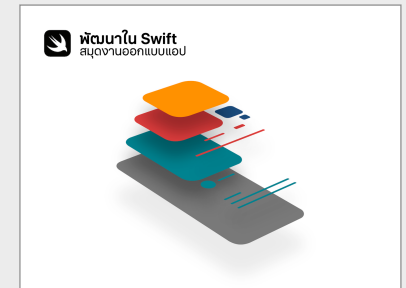
ในโมดูลนี้ สมาชิกชมรมจะใช้สมุดงานออกแบบแอป "พัฒนาใน Swift" ในการออกแบบแอปเพื่อช่วยแก้ปัญหาที่พวกเขาสนใจ สมาชิกชมรมจะใช้เฟรมเวิร์กสำหรับการคิดงานออกแบบเพื่อเรียนรู้การออกแบบแอป ซึ่งเป็นทักษะพื้นฐานของการพัฒนาแอป iOS พวกเขาจะได้ศึกษาเรื่องความสัมพันธ์ระหว่างการออกแบบแอปและการเขียนโค้ดใน Swift ผ่านแต่ละขั้นตอนของวงจรการออกแบบแอป เพื่อเปลี่ยนไอเดียแอปให้กลายเป็นจริง

จากนั้นคุณสามารถใช้คู่มือการแสดงผลงานแอปเพื่อช่วยสมาชิกชมรมวางแผนทำวิดีโอแนะนำแอปที่บันทึกกระบวนการทำงานและแสดงการออกแบบแอปของพวกเขา จากนั้นจัดแสดงผลงานแอปเพื่อยกย่องความช่างประดิษฐ์ของสมาชิกชมรม



วงจรการออกแบบแอป

แหล่งข้อมูล



[สมุดงานออกแบบแอป "พัฒนาใน Swift" >](#)



[คู่มือการแสดงผลงานแอป >](#)

ความท้าทายในการออกแบบแอป

สมาชิกชมรมตั้งทีมเล็กๆ ระบุปัญหาที่สนใจ จากนั้นออกแบบแอปเพื่อช่วยแก้ปัญหา เมื่อใช้สมุดงานออกแบบแอป "พัฒนาใน Swift" พวกเขาจะทำตามกระบวนการออกแบบของการกำหนดไอเดียแอป สร้างต้นแบบที่ใช้งานได้ใน Keynote รวมถึงทดสอบและประเมินแอปกับผู้ใช้งาน ก่อนจะนำผลที่ได้กลับไปปรับปรุงแอปของตัวเอง





© 2021 Apple Inc. สงวนลิขสิทธิ์ทุกประการ Apple, โลโก้ Apple, Apple TV, Apple Watch, iPad, iPhone, Keynote, Mac, MacBook Pro, macOS, Siri, Swift, Swift Playgrounds, โลโก้ Swift, watchOS และ Xcode เป็นเครื่องหมายการค้าของ Apple Inc. ซึ่งจดทะเบียนในสหรัฐอเมริกาและประเทศอื่นๆ tvOS เป็นเครื่องหมายการค้าของ Apple Inc. ส่วน App Store เป็นเครื่องหมายบริการของ Apple Inc. ซึ่งจดทะเบียนในสหรัฐอเมริกาและประเทศอื่นๆ iOS เป็นเครื่องหมายการค้าหรือเครื่องหมายการค้าจดทะเบียนของ Cisco ในสหรัฐอเมริกาและประเทศอื่นๆ และมีการใช้ภายใต้สิทธิ์การใช้งาน ชื่อสินค้าและบริษัทอื่นๆ ที่กล่าวถึง ณ ที่นี้อาจเป็นเครื่องหมายการค้าของบริษัทที่เกี่ยวข้อง เมษายน 2021