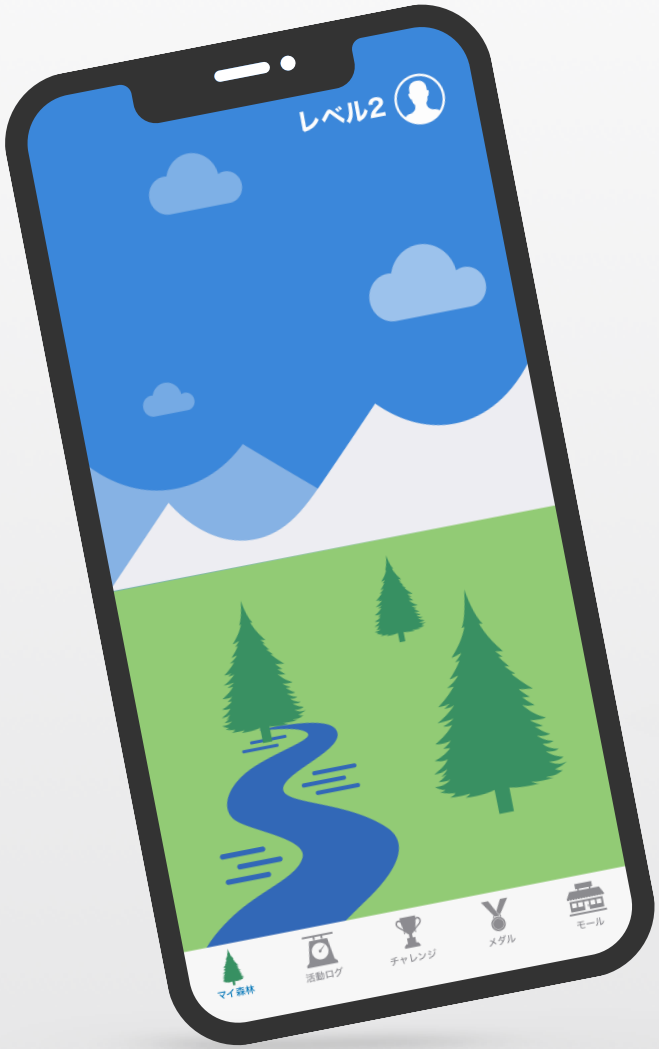
Develop in Swift

Swift Coding Club



Swift Coding Clubへようこそ

コーディングを学ぶと、創造力を発揮しながら協力し合って問題を解決する力が身につきます。また、自分のアイデアを形にすることを手助けしてくれます。

Swift Coding Clubは、コーディングとアプリケーションデザインを学ぶのに最適なプログラムです。Appleが開発したプログラミング言語であるSwiftを使ったアクティビティは、コードの書き方やアプリケーションのプロトタイプ（試作品）の作り方、コードと日常生活を関連づける思考の仕方を学びながら、チームで協力し合う構成になっています。

教師でなくても、コーディングに詳しくなくても、Swift Coding Clubの運営は可能です。教材は各自が自分のペースで学べるようになっており、指導する立場の人もメンバーと一緒に学びながら進めることができます。そして、地域でアプリケーションコンテストを行い、メンバーのアイデアやデザインを讃えましょう。

このガイドには3つのセクションがあります。

開始準備

Swift Coding Clubをスタートするために必要な準備について説明します。

指導計画と応用のヒント

活動を計画するためのモジュールやアクティビティの内容について紹介します。

作品発表

地域イベントを企画・開催するのに役立つ資料を紹介します。

コーディングに関するリソース

Swift Coding Clubでは、コーディングを学ぶための様々なリソースをご用意しています。これらのリソースを使えば、iPadでの基礎学習からMacでの本格的なアプリケーション開発へと学習を進めていくことができます。

Everyone Can Code | 10歳程度から

iPadまたはMacのSwift Playgroundsで Swiftを使ってプログラミングの基礎を学びます。
Everyone Can Codeの教材についてさらに詳しく >

Develop in Swift | 15歳程度から

MacのXcodeでアプリケーション開発を学びます。
Develop in Swiftの教材についてさらに詳しく >

開始準備

1. Develop in Swiftのリソースを詳しく確認する

Develop in Swiftの教材を活用して、MacでXcodeの使い方とSwiftを生徒たちに教えましょう。Swiftは、Appleが開発したパワフルで直感的なオープンソースのプログラミング言語で、急速に拡大しているアプリケーション経済圏で、プロのデベロッパがiOS、macOS、tvOS、watchOSなどのアプリケーション開発に使っているのと同じ言語です。コーディング初心者から経験者まで、すべてのメンバーのやる気を引き出すのに最適な言語でもあります。クラブ活動の企画を始める前に、Develop in Swiftに関する以下のリソースを確認しておくことをおすすめします。

Xcode

Xcodeは、プロのデベロッパが実際にアプリケーション開発に利用している統合開発環境です。ユーザーインターフェイス(UI)のデザインやコードの実装から、アプリケーションのテストとデバッグ、そしてApp Storeへの配布準備まで、アプリケーションを完成させるのに必要なツールが揃っています。

Xcodeをダウンロードして使ってみる >

Develop in Swift探究

コンピュータの主要な概念を学び、Swiftを使ってプログラミングスキルの土台をしっかりと作り上げていきます。iOSアプリケーションの開発に取り組みながら、コンピュータやアプリケーションが社会や経済、文化に与えた影響について学習していきます。レッスンでは、ブレーンストーミング、計画、プロトタイプ作成、評価の4段階で構成されるアプリケーションデザインプロセスを、オリジナルのアプリケーションの開発を通じて体験します。

「Develop in Swift」の教材をダウンロード >

2.テクノロジーを確認する

初回のミーティングまでに、必ず以下のものを用意してください。

- **Mac。**macOS Catalina以降を搭載したMacが必要です。1人1台ずつ用意するのが理想的ですが、何人かでデバイスを共有して一緒にコーディングに取り組むこともできます。
- **Xcode 11以降。**Appleが提供する無料のMacアプリケーションで、あらゆるMacアプリケーションやiOSアプリケーションの開発に使用されています。素晴らしいアプリケーション体験を生み出すために必要なあらゆるツールが揃っています。
- **「Develop in Swift探究」。**Appleが提供する無料のリソースです。初心者向けにコンピュータの主要な概念を紹介し、Swiftを使ってプログラミングスキルの土台をしっかりと作り上げていきます。
- **Keynote。**アプリケーションのプロトタイプを作るときにMacで使います。

Apple製品に関するご質問やお問い合わせは、Appleサポートのサイトにアクセスしてください。

3.計画を立てる

次の点を考慮してください。

- クラブにはどのようなメンバーが所属していますか？
  メンバーは何に興味をもっていますか？
  コーディングの経験はありますか？
  それとも完全な初心者ですか？
- どのくらいの頻度でクラブ活動を行いますか？
  コーディング活動に何時間確保できますか？
- クラブではどのようなテクノロジー環境を利用できますか？
- クラブの目標は何ですか？

4.情報発信する

Swift Coding Clubのことを多くの人に知らせましょう。クラブの新メンバーを募集するためのアイデアや資料を以下に紹介します。

- **クラブ発足の告知。**Eメール、ソーシャルメディア、ウェブサイト、チラシ、口コミなどを利用して、地域の人たちにクラブを紹介します。

- **説明会を開く。**クラブへの参加を考えている人に、興味のあることや、取り組んでみたいプロジェクトの種類などを聞いてみましょう。地域イベントの開催予定や、参加方法についても説明しましょう。クラブの簡単な紹介ビデオをオンラインで共有するのもよいでしょう。

Swift Coding Clubの宣伝やカスタマイズには、以下のアイテムも活用してください。

- **ポスター。**無料テンプレートをダウンロードし、カスタマイズして独自のポスターを作成してください。印刷して掲示するか、デジタル版ポスターをオンラインで見られるようにするとよいでしょう。クラブ活動の日程や場所、参加方法などの詳細も必ず記載しましょう。

- **ステッカーとTシャツ。**Swift Coding Clubステッカーをクラブの宣伝に利用してください。Tシャツもおすすめです。おそろいのTシャツを着れば、アプリケーションコンテストのときにクラブのメンバーを見分けやすくなります。Swift Coding ClubのTシャツテンプレートをダウンロードして、メンバー用のTシャツを作ってください。

Swift Coding Clubのポスター

Swift Coding Clubのステッカー

Swift Coding ClubのTシャツ

クラブ指導者のためのヒント

リーダーチームを編成する。クラブの指導を手伝ってくれるメンバーを選んでリーダーチームを作っておくと、活動が円滑になり、さらに盛り上がります。どのメンバーがリーダーに向いていそうですか？イベント、コーディング、アプリケーションデザインなどをそれぞれ担当する係を決めておくのもよいでしょう。

一緒に学ぶ。クラブの指導者が何もかも知っている必要はありません。メンバーが自分で調べたり問題解決するスキルを伸ばせるようサポートし、お互いに助け合って活動するよう促しましょう。

披露する。アプリケーションコンテストは、クラブの活動内容や、アプリケーションのアイデア、コーディングスキルを、メンバーの友達、家族、教師、地域の人たちにアピールする絶好のチャンスです。これをきっかけに新しいメンバーが増えるかもしれません。コンテストを開催するためのヒントについては、12ページを参照してください。

アイデアを共有する。ゲームを作りたいメンバーもいれば、人の役に立つアプリケーションを作りたいメンバー、Swiftについて学びたいメンバー、ロボットを動かしたいメンバーもいるでしょう。みんなが関心をもって一緒に取り組めるプロジェクトにするためには、どうしたらいいのかを考えましょう。

混合チームを結成する。先に進んでいるメンバーが、ほかのメンバーを置いてけぼりにしてしまうこともあります。進みの早いメンバーと初心者メンバーをペアにして、ペアプログラミングを試してみましょう。誰かに教えることで、理解も深まります。

指導計画と応用のヒント

1. Swiftについて

Swiftは、アプリケーションを開発するためにAppleが作った、直感的に使えるパワフルなプログラミング言語です。これは、急成長を続けるアプリケーション開発の世界において、プロのデベロッパがiPad、Mac、Apple TV、Apple Watch用のアプリケーション開発に使用しているものと同じ言語です。Swiftを使えば、プログラミングがかつてないほど簡単で自由に、そしてもっと楽しくなります。

Swiftについて詳しくは、swift.org（英語）にアクセスしてください。

2. Xcodeと「Develop in Swift探究」ガイドを使う

クラブの教材は、Xcodeのアプリケーションプロジェクトを中心に作られています。Xcodeは、プロのデベロッパが実際にアプリケーション開発に利用している統合開発環境です。コードの記述や管理に使用するソースコードエディタ、問題の診断に使用するデバッガ、そしてアプリケーションのビジュアル要素を配置し、コードと結びつけて実装できるInterface Builderというユーザーインターフェイスエディタが含まれています。

Xcodeについて詳しくは、Xcodeのサポートページをご覧ください。

「Develop in Swift探究」ガイドでは、XcodeのPlaygroundアクティビティに取り組みながら、コーディングの基礎を学びます。XcodeのPlaygroundでSwiftのコードを記述すると、ライブビューですぐ結果を確認することができます。コードでいろいろと試し、どのように動くか確認することで、初歩的なコーディングスキルを身に付けて新しいアイデアを試せるようになります。

「Develop in Swift探究：教師用ガイド」には追加のアクティビティが用意されています。これらを活用することで、メンバーの関心を高め、理解を深めることができるほか、継続して取り組むようにやる気を引き出すことができます。

「Develop in Swift探究：教師用ガイド」をダウンロード >

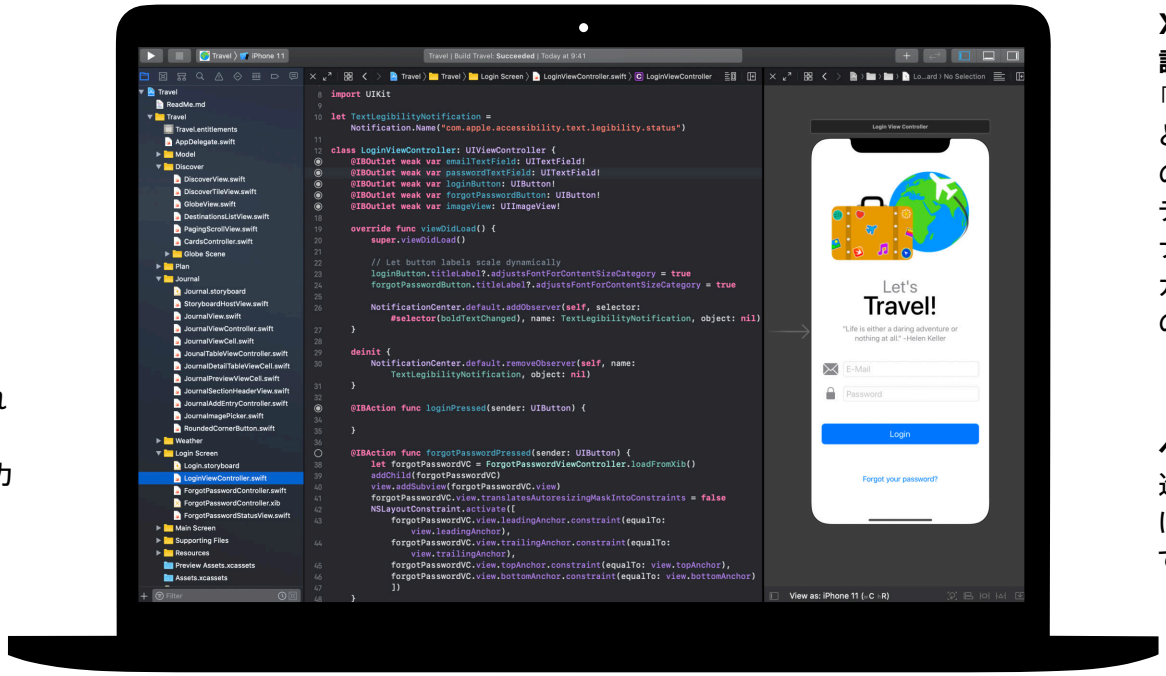
Xcodeアプリケーションをダウンロード >

教師用ガイドをダウンロード >

Xcodeで学習する際のヒント

コードの書き方は人それぞれ。
お互いにコードを見せ合って気付いたことを指摘したり、協力してデバッグしたりするとよいでしょう。

デバッグツールを使う。
アプリケーションがクラッシュすると、コード内でエラーが発生した行が赤くハイライト表示されます。コード内でprint()を使い、関連情報をコンソールにログ出力しましょう。ブレークポイントを設定すると、アプリケーションの実行を一時的に停止させ、変数を確認しながら、コードを1行ずつステップ実行できます。



Xcodeの環境設定を詳しく調べる。
メニューバーで「Xcode」>「Preferences」と選択して、テキスト編集などの環境設定を設定します。デベロッパアカウントの追加、ナビゲーションやフォントのカスタマイズ、イベント発生時の動作設定などが可能です。

ヘルプデスクを作る。
進みの早いメンバーが、ほかのメンバーをサポートできる場所を確保します。

手を止めて考える。
バグは避けられないものです。手を止めて、問題について考えてみましょう。どのような症状が発生しましたか？どの時点まで正常に動いていましたか？

学びを継続する。
探究コースを完了したメンバーは「Develop in Swift Fundamentals」、「Develop in Swift Data Collection」コース（英語）に進み、より高度で複雑なアプリケーションの開発に取り組みながら、知識やスキルをさらに磨いていくことができます。

キーボードショートカットをマスターする。
プロジェクトをビルドして実行する：⌘＋R
選択したコードをコメントアウトする／コメントアウトを外す：⌘＋/
選択したコードをインデントし直す：^＋I
インスペクタを表示する：⌘＋⌥＋0
ドキュメントを表示する：⌘＋⇧＋0

3.プロジェクトを選ぶ

クラブの教材は、アプリケーションプロジェクトのモジュールとアプリケーションデザインチャレンジが中心の構成となっています。メンバーは、Xcodeを使って一連のPlaygroundアクティビティや、解説付きのアプリケーション開発プロジェクトに取り組みながら、プログラミングの概念を学びます。「Develop in Swift探究」のコースには、各モジュールの実施に必要なすべてのことが用意されています。

冒頭にあるいくつかのアプリケーションプロジェクトには予備知識は必要ありません。後続のプロジェクトでは、難易度が徐々に高くなります。各プロジェクトの難易度を確認し、担当するメンバーのコーディング経験に合わせて、最初に取り組むプロジェクトを選んでください。

アプリケーションデザインチャレンジは、別のモジュールと並行して実施することも、単独の課題として取り組むこともできます。

モジュール1：PhotoFrameアプリケーション

モジュール2：QuestionBotアプリケーション

モジュール3：ColorMixアプリケーション

モジュール4：ElementQuizアプリケーション

モジュール5：アプリケーションデザインチャレンジ

コースをダウンロード >

4.アプリケーションデザインチャレンジ

Xcodeでアプリケーションを作る方法を学ぶのと並行して、メンバーは作ってみたいアプリケーションの構想を立て始めて、アイデアを共有し、プロトタイプを作成し、ほかのメンバーとそのアプリケーションをテストして、ユーザー体験を作り上げていくことができます。アプリケーションデザインチャレンジでは、学んだプログラミングスキルを発揮しながら、想像力豊かに工夫して取り組むことができます。

メンバーは優れたアプリケーションの機能や、自分のアプリケーションをデザインする上で検討すべき内容について学びます。「アプリケーションデザインジャーナル」を進めてアプリケーションデザインプロセスに取り組み、実際に動かせるプロトタイプを作成し、アプリケーションコンテストで紹介します。メンバーには、各セッションの前半でアプリケーションプロジェクトに取り組んだ後、残りの時間で独自のアプリケーションアイデアに取り組んでもらいます。または、アプリケーションプロジェクトのセッションと独自のアプリケーションデザインのセッションを交互に実施してもよいでしょう。

5. さらに進める

メンバーの興味に合わせてセッションを追加することもできます。デザインとコーディングのアクティビティを発展させ、接続デバイス用のアプリケーションやwatchOSアプリケーションに取り組んだり、機械学習や拡張現実などのトピックをさらに掘り下げて、未来のアプリケーションデザインについて調べてもよいでしょう。

デザインのブレーンストーミングを盛り上げるため、ゲストスピーカーを招待したり、校外活動を行うのもよいでしょう。プロジェクトの対象ユーザーやデザイン要件に関するメンバーの理解が深まります。

作品発表

地域のイベントやリモート形式のアプリケーションコンテスト

地域でのイベントやリモート形式のアプリケーションコンテストの開催は、地域の幅広いメンバーを巻き込んで、現代の社会問題を解決するというコーディングの可能性について一緒に考える絶好のチャンスです。また、クラブのメンバーたちの才能を披露する、またとない機会でもあります。

1. 盛大なイベントを企画する。日程を決め、生徒、教師、家族、地域のメンバーを招待しましょう。

実際に集まるイベントまたはリモート形式で、チームごとにプロジェクトを紹介する時間を設け、簡単な質疑応答を行います。クラブの規模が大きい場合は、前半と後半に分けて、メンバーがお互いのプレゼンテーションを見ることができるようにしてください。

クラブのセッション中に撮った写真を楽しいスライドショーにして、最後に上映するのもおすすめです。

2. 賞を設定する。楽しく競い合えば、学習意欲が高まります。コーディングやデザインで特に優秀だったメンバーを表彰して、やる気を刺激しましょう。次のような賞を検討します。

- 最優秀エンジニアリング賞
- 最優秀アイデア賞
- 最優秀デザイン賞
- 最優秀プレゼン賞

特別賞を追加し、観客に投票してもらうこともできます。

この賞状をダウンロードし、カスタマイズして様々な賞状を作ってください。

3. 審査員やメンターを依頼する。審査員やメンターには、教職員、コーディング経験のある生徒、教育委員会のメンバー、アプリケーション開発やデザイン業界のエキスパート、自治会長、アプリケーションプロジェクトの対象ユーザーなどが適任です。

審査員にはアプリケーションコンテストよりも前にクラブ活動を見学してもらうこともできます。審査員をゲストスピーカーとして招き、プロジェクトのブレーンストーミングや計画の段階で、プロの視点からの助言をお願いしてもいいでしょう。

4. シェアしてインスピレーションを広げる。プレゼンテーションを録画してもよいでしょう。ビデオを幅広いコミュニティで共有しましょう。また、ハイライトを紹介するバージョンを作成して、将来のメンバーにインスピレーションを与えるために活用してください。

Develop in Swift
Swift Coding Club

修了認定証

さん

あなたは

を修了したことを、ここに認定します。

サイン

日付

Swift Coding Clubのモジュール

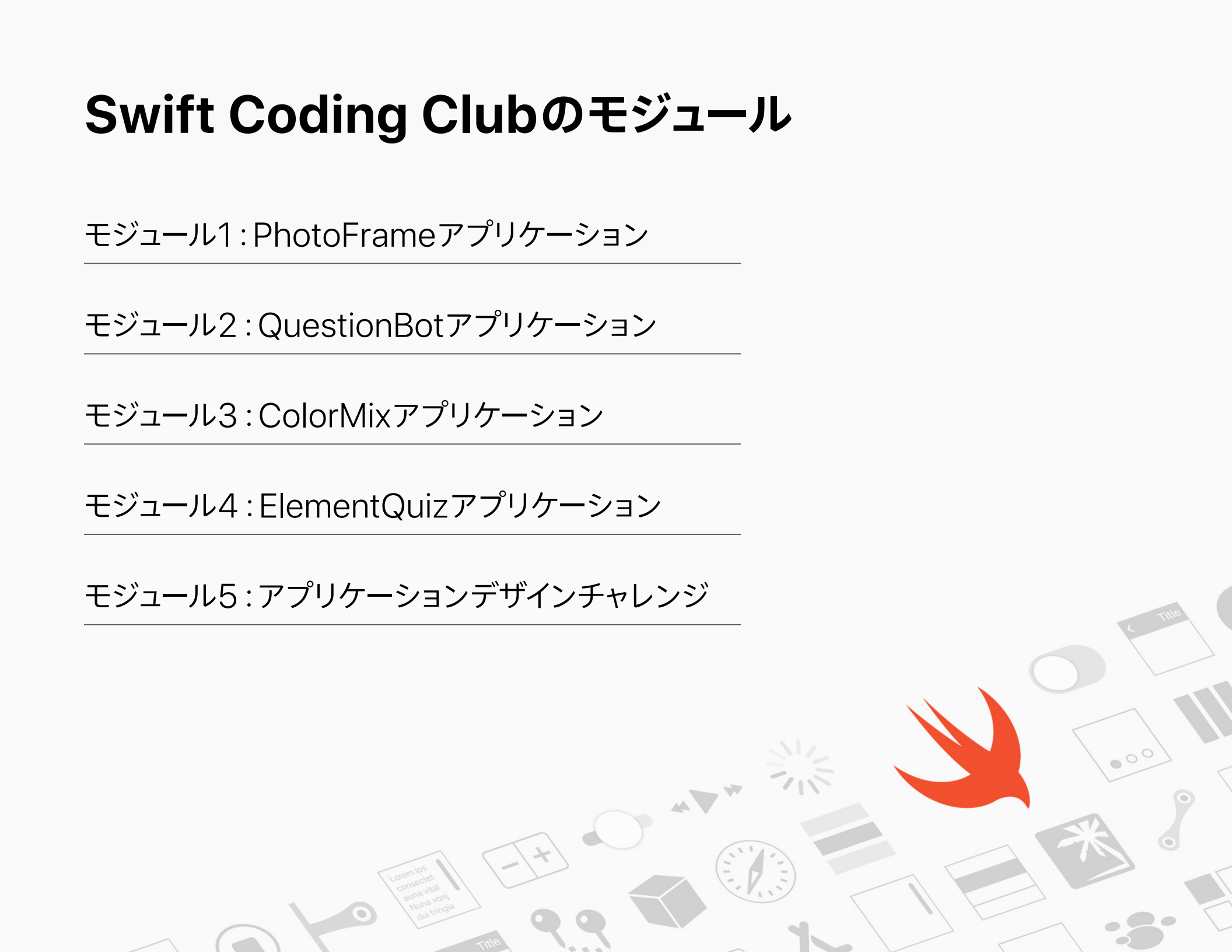
モジュール1 : PhotoFrameアプリケーション

モジュール2 : QuestionBotアプリケーション

モジュール3 : ColorMixアプリケーション

モジュール4 : ElementQuizアプリケーション

モジュール5 : アプリケーションデザインチャレンジ



PhotoFrameアプリケーション

モジュール1

PhotoFrameアプリケーション
モジュール1の概要

最初のアプリケーションは簡単に作れます。このモジュールでは、基本的なUIコンポーネント（写真など）を表示するアプリケーションの開発に必要な主要概念とスキルを学びます。UIコンポーネントの基本を理解することは、どのようなアプリケーションを作る場合でも重要になるため、この知識はコーディングスキルやアプリケーション開発スキルを伸ばしていく上でも役に立ちます。このプロジェクトでは、Xcode、Interface Builder、シミュレータの使い方に慣れながら、それらを組み合わせてアプリケーションを開発する方法を学びます。

セッション1〜7

値について学び、XcodeのPlaygroundで値、変数、定数を使う練習をします。

- Playgroundの基本
- 命名と識別子
- 定数と変数
- 文字列

セッション8〜9

学んだスキルと概念を応用して、単語ゲームのPlaygroundを作ります。

セッション10〜12

XcodeとInterface Builderを使ってPhotoFrameアプリケーションを作ります。

PhotoFrameアプリケーション

1 Xcodeの概要

XcodeのPlaygroundの操作方法を確認し、ベーシックなプログラムのコードを入力して変更する方法を学びます。

学ぶ：プログラミングの基本事項を確認し、データの入力と出力の役割について学びます。

プログラミングとは（15ページ）
値（16ページ）

練習する：コードを入力して変更する方法を学びます。

「Playground Basics」のPlayground（27〜29ページ）

2〜3 命名と識別子

プログラミングでの命名の重要性を学び、問題を解決する簡単なプログラムをいくつか作ります。

学ぶ：命名と識別子が重要となる理由を確認し、新しいゲームのデザインをスケッチしながら、主要なコンポーネントに名前を付けます。

命名と識別子（18〜19ページ）

練習する：簡単な問題を解決するプログラムを作りながら、命名する練習をします。

「Naming and Identifiers」のPlayground（30〜33ページ）

4〜5 定数と変数

変数と定数を宣言する方法を学び、得点を記録するプログラムを作ります。

学ぶ：変数と定数の違いを学び、自分たちの存在がプログラムだったとしたらどうなるか考えます。

定数と変数（20ページ）

練習する：ゲームの得点を記録するプログラムを作ります。

「Constants and Variables」のPlayground（38〜41ページ）

PhotoFrameアプリケーション

6〜7 文字列

文字列について学び、コードで文字列を使って簡単なゲームを作成する方法を確認します。

学ぶ：文字列の重要な特性を確認し、チャットボットのオリジナルの回答を作成します。

文字列（23〜24ページ）

練習する：空欄を埋めて文章を完成させるゲームを作ります。

「Strings」のPlayground（42〜44ページ）

8〜9 単語ゲーム

値、定数、文字列について学んだ知識を使って、Playgroundで単語ゲームを作成し、ほかのメンバーに挑戦してもらいます。

発展させる：単語を置き換えて面白いストーリーを作るゲームを作ります。

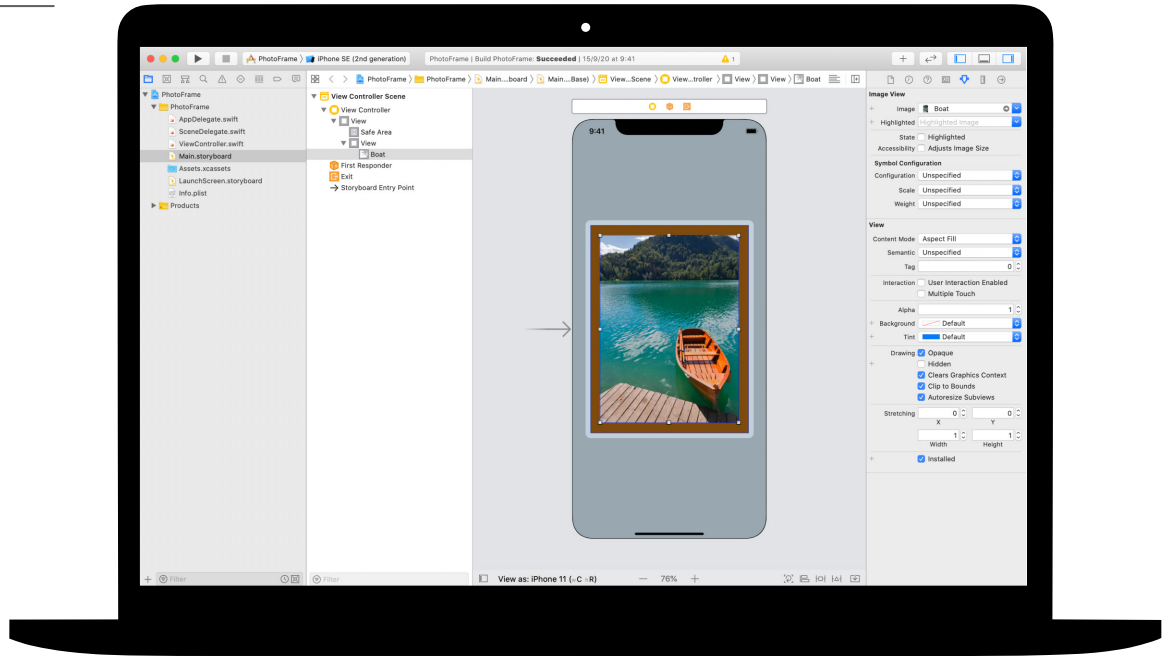
「Word Games」のPlayground（45〜46ページ）

10〜12 PhotoFrameアプリケーション

XcodeのInterface Builderの使い方を確認し、シンプルなアプリケーションを作って実行します。

発展させる：フレームをカスタマイズして写真を表示するアプリケーションを作って表示します。

PhotoFrameアプリケーションプロジェクト（47〜73ページ）

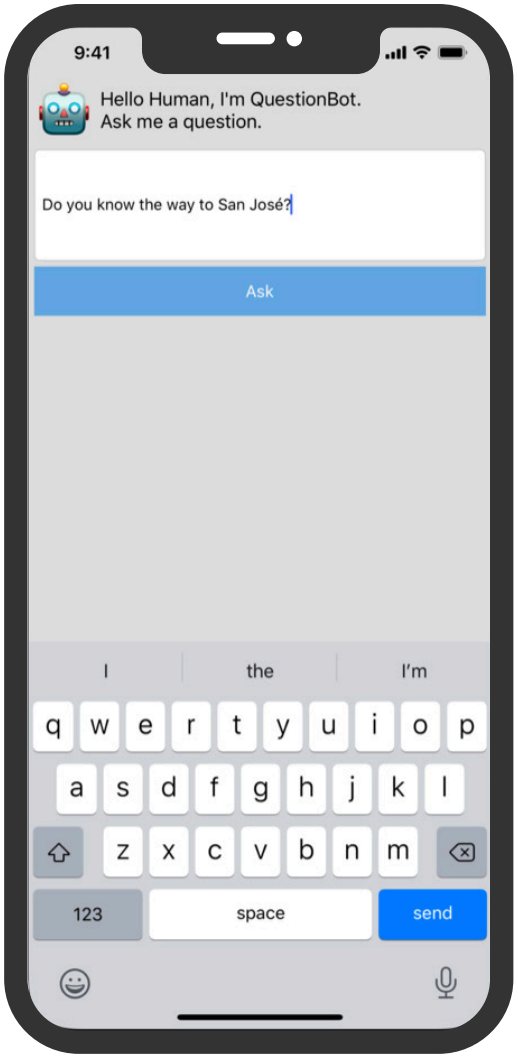


QuestionBotアプリケーション

モジュール2

QuestionBotアプリケーション
モジュール2の概要

クイズアプリケーションを使ったことはありますか？Siriがどのように動いているのか不思議に思ったことはありませんか？どのアプリケーションにも、その動作を定義しているロジックが組み込まれています。QuestionBotでは、質問に合わせて様々な応答をする「頭脳」部分を含むアプリケーションを作成します。そのために、アルゴリズムをデザインし、コードを関数にまとめ、様々な種類の型を使う方法などを学びます。このモジュールでは、アプリケーションの仕組みや、アプリケーションのインターフェイスを制御するコードの作成方法、そしてアプリケーションで人間の思考回路を再現するために利用できるロジックについて学びます。



セッション1〜10

プログラミングの土台となるアルゴリズムについて学び、XcodeのPlaygroundで関数、型、パラメータを使う練習をします。

- アルゴリズム
- 関数
- 型
- パラメータ
- ブール値を使って意思決定を行う

セッション11〜12

学んだスキルと概念を応用して、「BoogieBot」のPlaygroundでダンスルーチンを作成します。

セッション13〜14

Xcodeを使って、QuestionBotアプリケーションで質問に答える「頭脳」部分をプログラムする機能を追加します。

QuestionBotアプリケーション

1〜2 アルゴリズム

プログラミングで重要なツールとなる
アルゴリズムについて学び、日常的な問題を
解決するアルゴリズムを作成する練習をします。

学ぶ：アルゴリズムで「シーケンス」と「選択」を
使って簡単な問題を解決し、気分に合わせて
音楽を選ぶプログラムのアルゴリズムを作ります。

アルゴリズム（109ページ）
シーケンス（110ページ）
選択（111ページ）

3〜4 関数

再利用可能なコードのセクションを
作成できる関数について学び、作詞する
プログラムを作ります。

学ぶ：「夕食の準備」といった身近な例を使って、
命令を関数の中にまとめる練習をします。

関数（112〜114ページ）

練習する：同じフレーズを繰り返す歌を作成する
プログラムを作ります。

「Functions」のPlayground（121〜124ページ）

5〜6 型

様々な種類のデータを区別できる型について
学び、簡単な計算を実行できるプログラムを
作ります。

学ぶ：型を使って値を記述する方法を学び、
おもちゃの組み立てキットに使う部品の種類（型）
について考えます。

型（115〜116ページ）

練習する：簡単な計算を実行するプログラムを作
ります。

「Types」のPlayground（125〜127ページ）

QuestionBotアプリケーション

7〜8 パラメータ

パラメータを使って関数への入力を定義する方法を学び、入力値に応じて異なる文章を出力するプログラムを作ります。

学ぶ：パラメータを使ってより柔軟な関数を作り、細かいニーズに対応できるようにタ食を準備する関数を改良します。

パラメータ（116〜117ページ）

練習する：関数を使って、渡した値に応じて異なる文章を出力するプログラムを作ります。

「Parameters and Results」のPlayground（128〜130ページ）

9〜10 ブール値を使って意思決定を行う

プログラミングでブール型を使ってできる様々なことをを学び、特定の年がうるう年かどうか判断するプログラムを作ります。

学ぶ：ブール型について学び、困っているロボットを手助けするのに使います。

ブール値を使って意思決定を行う（118ページ）

練習する：うるう年かどうか判断するプログラムを作ります。

「Making Decisions」のPlayground（131〜134ページ）

11〜12 BoogieBot

関数を使って細かいパーツで構成される複雑なダンスルーチンを作り、独自の振り付けをアニメーション画像としてほかの人に共有します。

発展させる：BoogieBotのダンスルーチンを作成し、完成した作品をアニメーション画像として保存します。

「BoogieBot」のPlayground（135ページ）

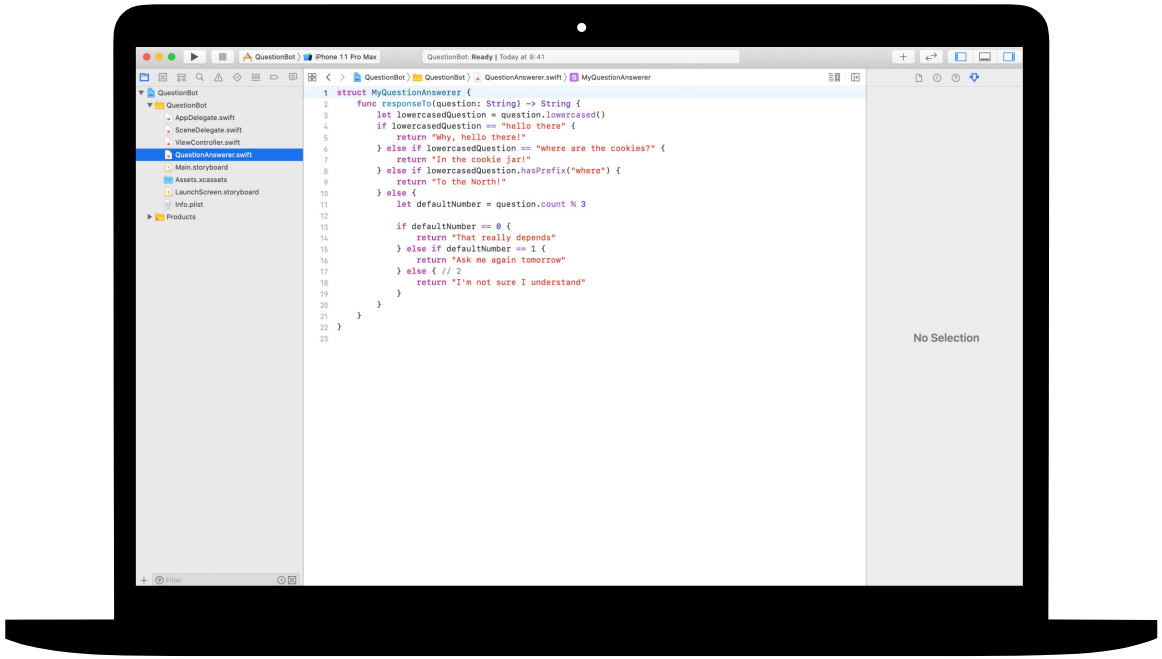
QuestionBotアプリケーション

13〜14 QuestionBot

QuestionBotアプリケーションのロジックを作り、質問に応じて異なる応答を返すようにします。

発展させる：QuestionBotアプリケーションの「頭脳」部分をプログラムして質問にどのように回答するかを決め、コードをテストしてトラブルシューティングを行う方法を学びます。

QuestionBotアプリケーションプロジェクト
(137〜150ページ)

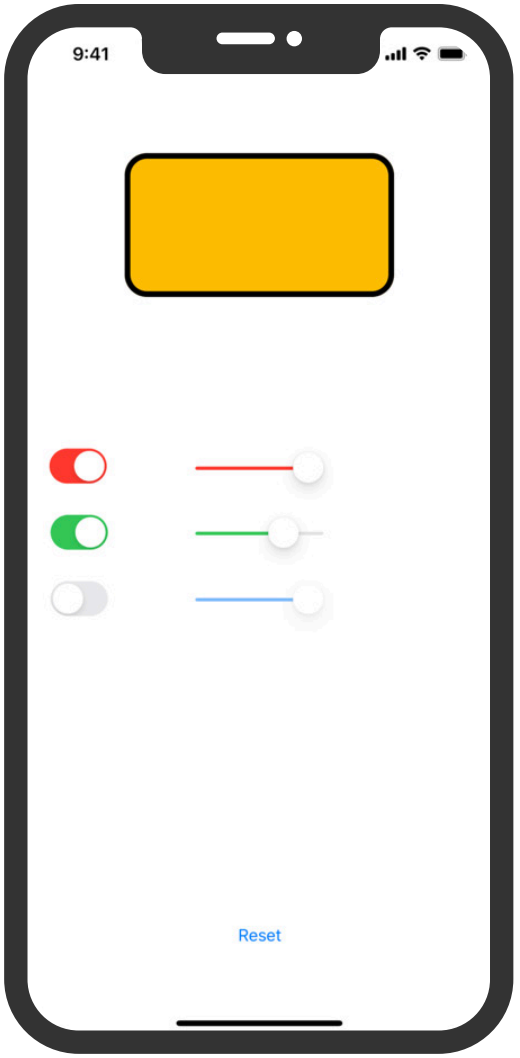


ColorMixアプリケーション

モジュール3

ColorMixアプリケーション
モジュール3の概要

iPhoneのユーザーインターフェイス(UI)について考えましょう。ここまで、ベーシックなUI要素を使ってアプリケーションを作り、UIを制御するロジックを作る方法を学んできました。ColorMixでは、ボタンやスイッチといったコントロールを備えたインタラクティブなアプリケーションを作る方法を学びます。
さらに、これらの視覚的なUI要素をSwiftコードに接続し、目的に合わせて動作させる方法も学習します。そのために、プロパティとメソッドを持つ独自のカスタム型を定義し、型のインスタンスを使って、データを配列で収集する方法を学びます。最終的には、赤、緑、青を混ぜて虹のすべての色(またはそれ以上の色)を生成する、ColorMixというアプリケーションができあがります。



セッション1〜6

データを整理する方法を学び、カスタム型のメソッドやプロパティを定義する練習をして、XcodeのPlaygroundで配列を操作します。

- インスタンス、メソッド、プロパティ
- 配列とループ
- 構造体

セッション7〜8

グラフィックスを作る仕組みを学び、グラフィックスや絵文字、線画アニメーションを1ピクセルずつ描画して作成します。

セッション9〜12

UIにスイッチやスライダを追加して、ColorMixアプリケーションを作ります。

ColorMixアプリケーション

1〜2 インスタンス、メソッド、プロパティ

型のインスタンスを作成して、そのメソッドとプロパティを使う方法を学び、ダンス対決するロボットをプログラムします。

学ぶ：型でメソッドとプロパティを定義する方法を学び、様々な種類の動物のメソッドとプロパティを記述します。

インスタンス、メソッド、プロパティ（184ページ）

練習する：2体のロボットのダンス対決を行うプログラムを作ります。

「Instances, Methods, and Properties」のPlayground（196〜198ページ）

3〜4 配列とループ

配列内のデータを並べ替える方法とループを使って配列を処理する方法を学習し、票の集計、進捗状況の追跡、キーワードの検索を行うプログラムを作ります。

学ぶ：アルゴリズムで反復処理を行い、ループを使って配列内の要素を操作します。ボードゲームのゲーム内容を記述するアルゴリズムを作り、コレクションを操作する方法について考えます。

リストと配列（185ページ）
アルゴリズム：反復（186〜187ページ）
ループ（188ページ）
配列の処理 — 探索（189〜190ページ）

練習する：票の集計、毎日の目標に対する進捗状況の追跡、キーワードによるメッセージのフィルタリングを行うプログラムを作ります。

「Arrays and Loops」のPlayground（199〜202ページ）

5〜6 構造体

構造体でカスタムの型を作成する方法を学習し、カスタムの型を使ってプログラミング上の課題を解決します。

学ぶ：構造体を使って独自の型を定義し、好きな動物のカスタム型を作ります。

構造体（Struct）で独自の型を定義する（191〜192ページ）

練習する：カスタムの型を使って問題を解決するプログラムを作ります。

「Structures」のPlayground（203〜205ページ）

ColorMixアプリケーション

7〜8 ピクセルアート

グラフィックスを作る仕組みを学び、1ピクセルずつ描画してオリジナルのグラフィックスを作成します。

発展させる：オリジナルのグラフィックス、絵文字、線画アニメーションを作るコードを作成します。

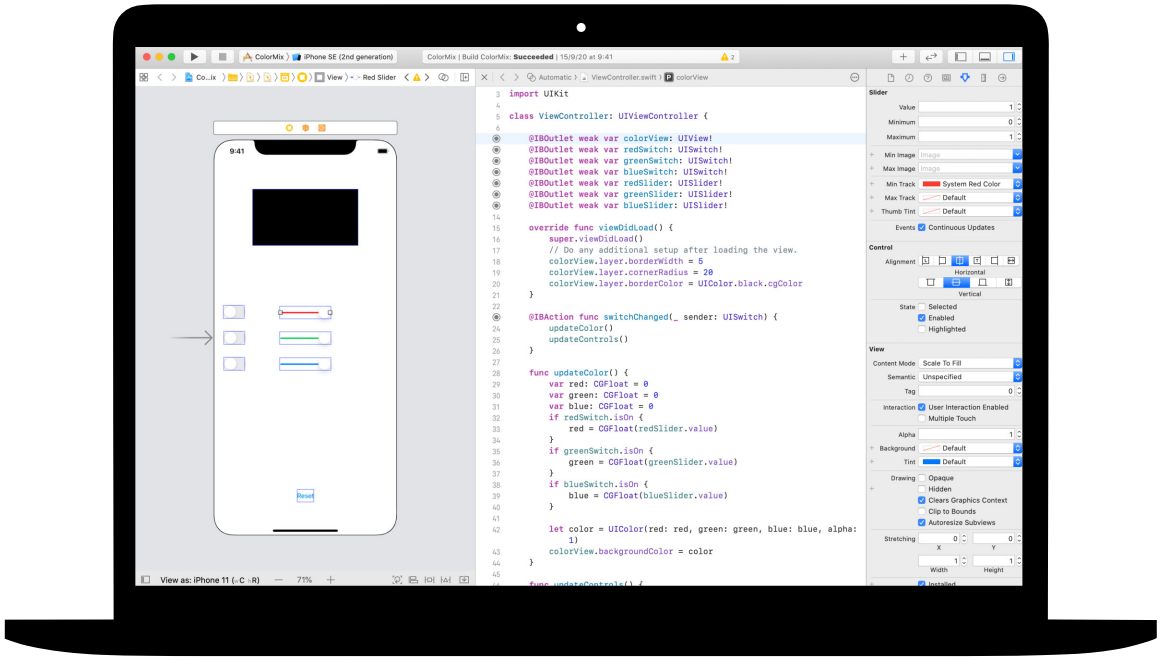
「Pixel Art」のPlayground（215〜216ページ）

9〜12 カラーピッカー

アクションとアウトレットを使って、Swiftのコードをアプリケーションのユーザインタフェースに接続する方法を学びます。

発展させる：スイッチ、スライダ、ボタンを使って、独自の色に混ぜるアプリケーションを作ります。

ColorMixアプリケーションプロジェクト（302〜345ページ）

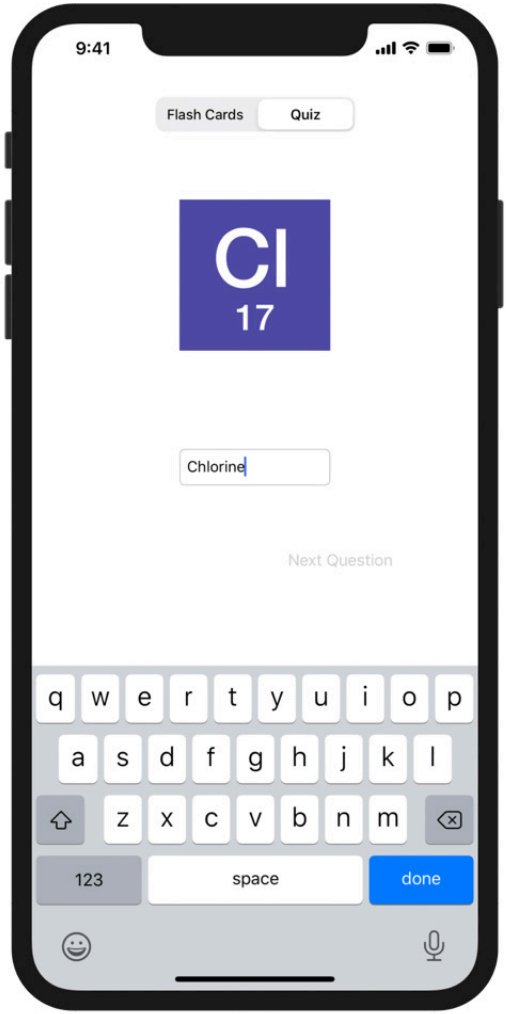


ElementQuizアプリケーション

モジュール4

ElementQuizアプリケーション
モジュール4の概要

多くの人は、整理整頓する、お金を管理する、移動経路を調べるなど、特定の問題を解決する目的でアプリケーションを使います。ElementQuizでは、元素周期表を覚えるのに役立つアプリケーションを作ります。列挙型について学習してから、アプリケーションの課題を1つ選び、学んだ内容を応用して個人で取り組みます。ミームメーカーアプリケーションやジャンケンゲームを作ったり、ElementQuizアプリケーションの改良に取り組んだりできます。



セッション1〜4

手順に従い、ElementQuizアプリケーションでフラッシュカード式のインターフェイスを作ります。

セッション5〜6

列挙型について学習し、投票を数えるプログラムを作ります。

- 列挙型とswitch文

セッション7〜12

3つのアプリケーションプロジェクトから1つ選んで取り組みます。

ElementQuizアプリケーション

1〜4 ElementQuiz アプリケーション：パートA

周期表の元素を覚えるのに役立つフラッシュカードアプリケーションを作る方法を学びます。

発展させる：元素記号の学習に役立つ、フラッシュカードインターフェイスを備えたクイズアプリケーションを作ります。

ElementQuizアプリケーションプロジェクト、パート1〜3（401〜416ページ）

5〜6 列挙型とswitch文

列挙型について学習し、票を数えるプログラムを作ります。

練習する：投票結果を集計するプログラムを作ります。

「Enums and Switch」のPlayground（206〜208ページ）

ElementQuizアプリケーション

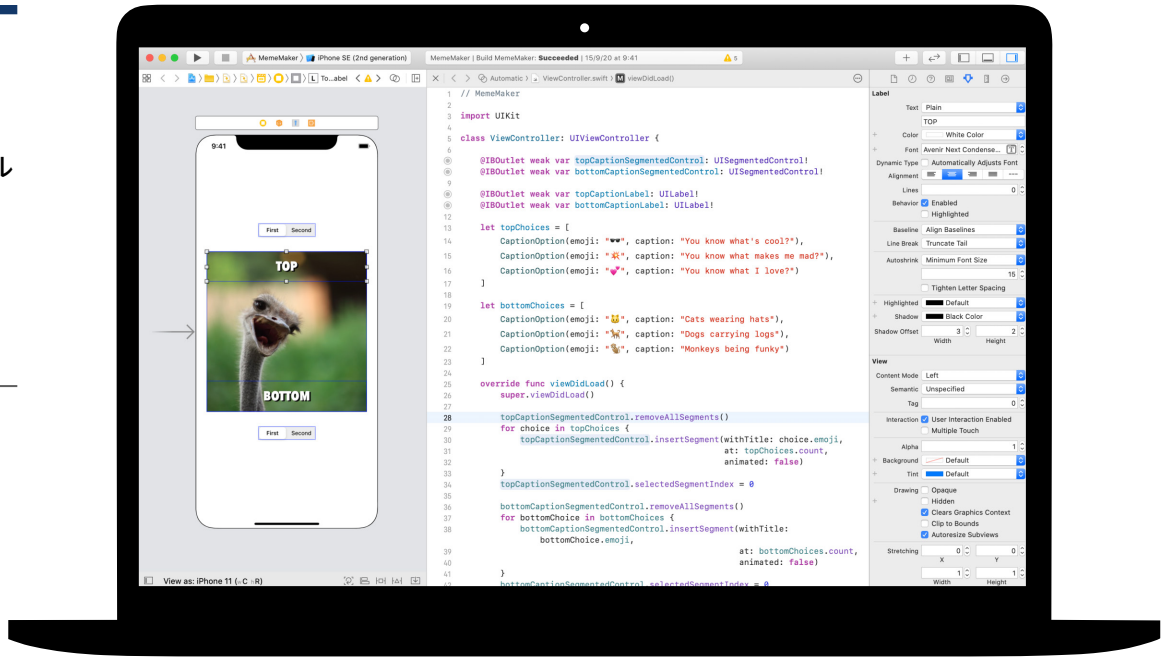
セッション7〜12では、以下の3つのアプリケーションプロジェクトから1つ選びます。星の数は難易度を表しています。

7〜12 ミームメーカーアプリケーション ★

セグメントコントロールを使って、画像の上部または下部に様々なキャプションを表示する方法を学びます。コントロールは独立しているので、テキストを組み合わせてカスタマイズすることができます。ジェスチャリコグナイザを使って、ユーザーが画面内でキャプションをドラッグできるようにする方法を学びます。

発展させる：気分に合わせて写真に楽しいカスタムキャプションを追加できるアプリケーションを作ります。

ミームメーカーアプリケーションプロジェクト
(384〜399ページ)



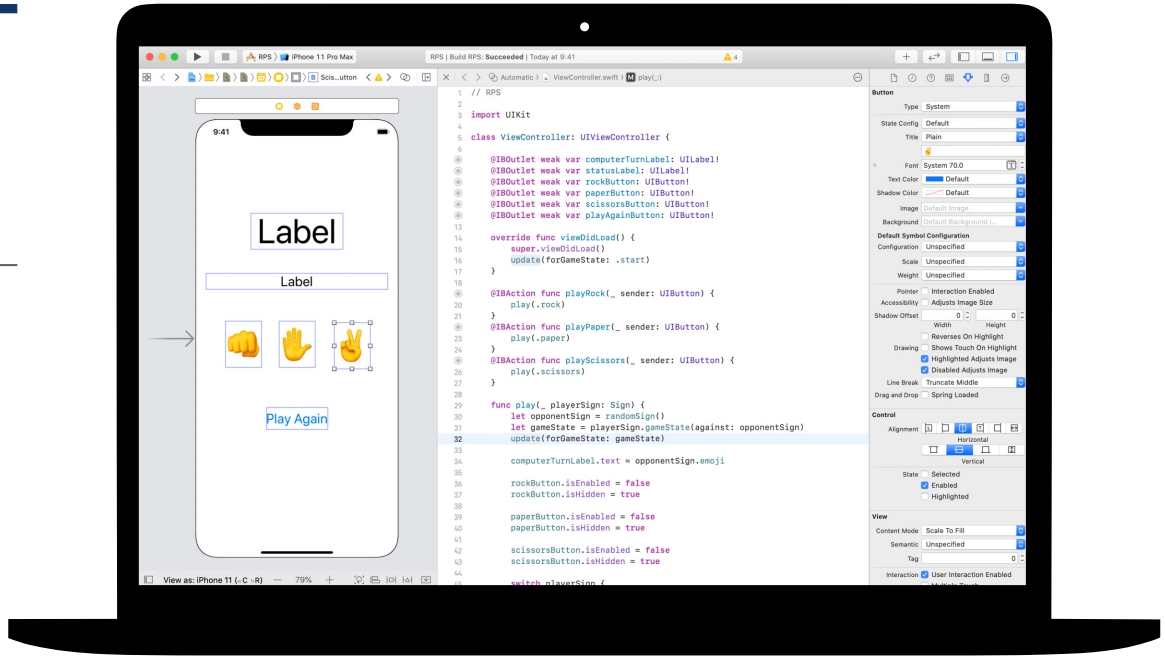
ElementQuizアプリケーション

7〜12 ジャンケンアプリケーション ★★

構造体と列挙型を使ってジャンケンゲームのモデルとロジックを作り、乱数を使ってコンピュータを相手にエンドレスで遊べるようにする方法を学びます。

発展させる：絵文字とボタンを使うゲームを作ります。

ジャンケンアプリケーションプロジェクト
(368〜383ページ)



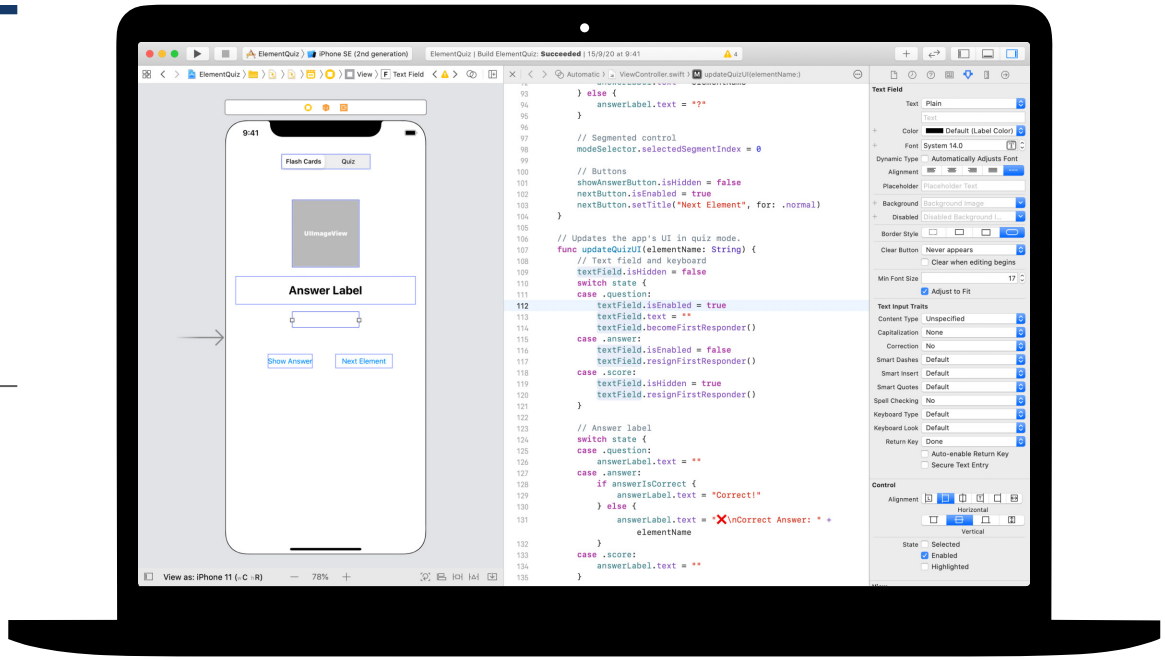
ElementQuizアプリケーション

7〜12 ElementQuizアプリケーション：パートB ★★★

ElementQuizアプリケーションで採点クイズモードを作成して、テキスト入力を処理する方法を学びます。ユーザーインターフェイスのロジックを組み立てる方法や、コードが複雑になった場合にリファクタリングする方法を学びます。

発展させる：クイズアプリケーションを発展させて、採点クイズモードを実装します。

ElementQuizアプリケーションプロジェクト、パート4〜10（416〜467ページ）



アプリケーションデザインチャレンジ

モジュール5

アプリケーションデザインチャレンジ
モジュール5の概要

このモジュールでは、メンバーは少人数のグループに分かれて地域社会の問題を解決できるアプリケーションをデザインします。デザインプロセス（ブレーンストーミングでアイデアを出す、アプリケーションを計画する、Keynoteで実際に動くプロトタイプを作成する、アプリケーションを評価する）を順に学びます。その後、各チームでアプリケーションのプロモーションビデオを作成します。作成過程のドキュメントを交え、アプリケーションをアピールしてもらいます。

メンバーは、デザインサイクルを進めながら「アプリケーションデザインジャーナル」にアイデアを書きとめ、後で確認することができます。デザインプロセスを記録しておくことで、アプリケーションプロジェクトの内容を繰り返し改善できるようになります。また、今後新しいプロジェクトを作成する時にテンプレートとして参照することができます。

このモジュールの最後にアプリケーションコンテストを開催し、メンバーの創意工夫を讃えましょう。

セッションの概要

- ブレーンストーミング：3セッション
- 計画：2セッション
- プロトタイプ：4セッション
- 評価：2セッション
- 発表：1セッション
- アプリケーションコンテスト

リソース

アプリケーションデザインジャーナル

アプリケーションデザインチャレンジ

1〜3 ブレーンストーミング

アプリケーションのアイデアを出し合って検討し、その目的や対象ユーザー、核となる機能を決めます。

ブレーンストーミング

- 目的
- アイデア
- 対象者
- 絞り込み
- 改善

4〜5 計画

アプリケーション内でiOSの機能をどのように利用するのかについて考え、アプリケーションのユーザーインターフェイス（UI）の中心となるデザイン要素を調べます。

計画

- UI／UX
- iOSの機能
- デザイン

6〜9 プロトタイプ

アプリケーションのUIをデザインし、画面のフローチャートを作り、実際に動くアプリケーションのプロトタイプをKeynoteで作成します。

プロトタイプ

- デザイン
- フローチャート
- 作成

アプリケーションデザインチャレンジ

10〜11 評価

仲間や地域の人とプロトタイプをテストし、
フィードバックに応じてデザインを修正する
手順を繰り返します。

評価

- 観察
- インタビュー

12 アプリケーションの発表

3分間のプレゼンテーションまたはビデオを
作成し、アプリケーションで解決しようとしている
問題と、その解決方法を説明します。

アプリケーションコンテスト

アプリケーションコンテストを開催し、クラブで作成したアプリケーションのプロトタイプを
より多くの人と共有しましょう。イベントの計画や開催についてヒントが必要な時は、
「アプリケーションコンテスト開催ガイド」を参照してください。