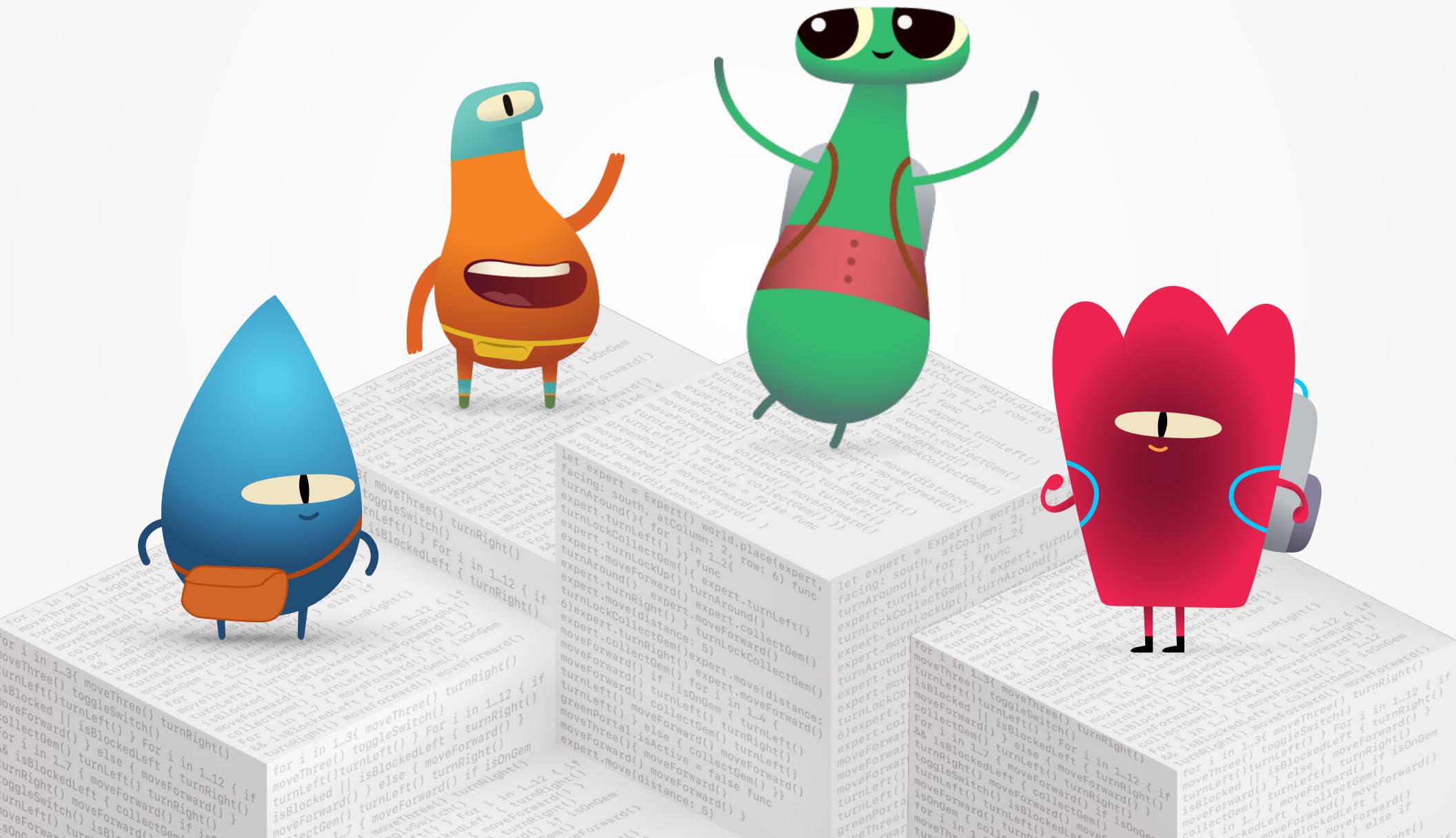




# Swift Coding Club

# Everyone Can Code



# Swift Coding Clubへようこそ

プログラミングを学ぶと、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では、プログラミングとアプリケーションデザインについて楽しく学べます。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、プログラミング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、プログラミ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讚えましょう。

このキットには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。

## プログラミングに関するリソース

Swift Coding Clubでは、プログラミングを教える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のSwift PlaygroundsでSwiftを使ってプログラミングの基礎を学びます。[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。[さらに詳しく >](#)

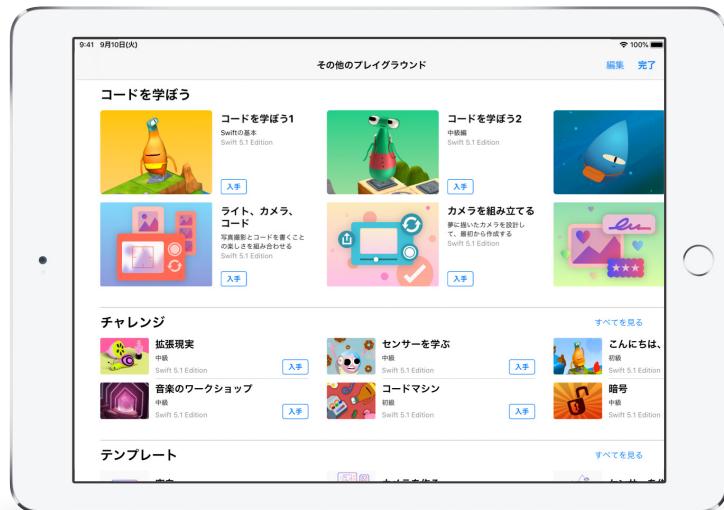


# 開始準備

## 1. Everyone Can Codeの教材を確認する

Everyone Can Codeでは、インタラクティブなパズルやゆかいなキャラクター、楽しいアクティビティを通して、プログラミングの世界に足を踏み入れることができます。クラブ活動の企画を始める前に、Everyone Can Code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。

Swift Playgroundsは、Swiftをインタラクティブに楽しく学べる無料のiPadアプリケーションです。内蔵されているレッスンのライブラリに加え、業界をリードするデベロッパや出版者が作成した追加の課題も用意されています。



[Swift Playgroundsをダウンロードして使ってみる >](#)

「Everyone Can Codeパズル」には、プログラミングの概念を学び、それらを日常生活に結びつけて考え、Swift Playgroundsのパズルを解きながらそれらの概念を応用するためのアクティビティが用意されています。



[Everyone Can Codeの教材についてさらに詳しく >](#)





## 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。

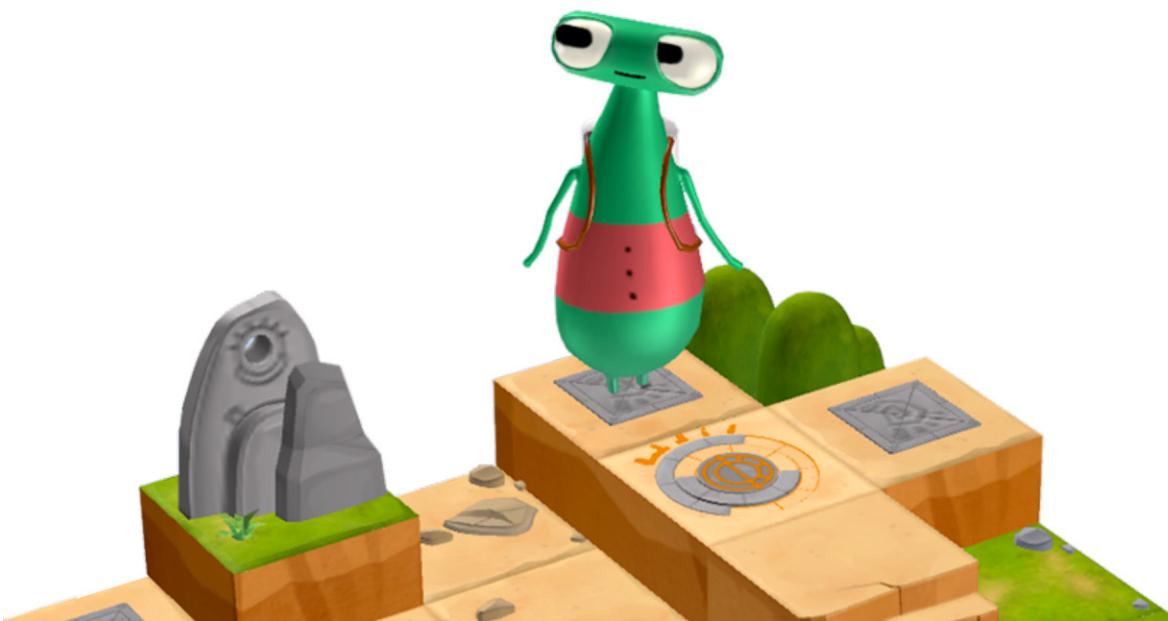
- **iPad。** Swift Playgroundsには、iOS 10以降またはiPadOS 13を搭載する64ビットのiPadが必要です。これにはiPad mini 2以降、iPad Air以降、iPad Proも含まれます。1人1台ずつ用意するのが理想的ですが、何人かでデバイスを共有して一緒にプログラミングを取り組むこともできます。
- **Swift Playgroundsアプリケーション。** [Swift Playgroundsをダウンロード >](#)
- **Everyone Can Codeパズル。** このブックを使って、「プロジェクトを作成する」および「クイズを作る」モジュールのアクティビティを進めます。 [Everyone Can Codeパズルをダウンロード >](#)

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？プログラミ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？夏合宿やイベントを計画している場合、プログラミングアクティビティ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。

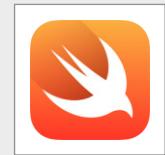
- **クラブ発足の告知。** Eメール、ソーシャルメディア、ウェブサイト、チラシ、口コミなどを利用して、地域社会にクラブの紹介をします。
- **説明会を開く。**クラブへの参加を考えている人に、興味のあることや、取り組んでみたいプロジェクトの種類などを聞いてみましょう。地域イベントの開催予定や、参加方法についても説明しましょう。クラブの簡単な紹介ビデオをオンラインで共有するのもよいでしょう。

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します。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

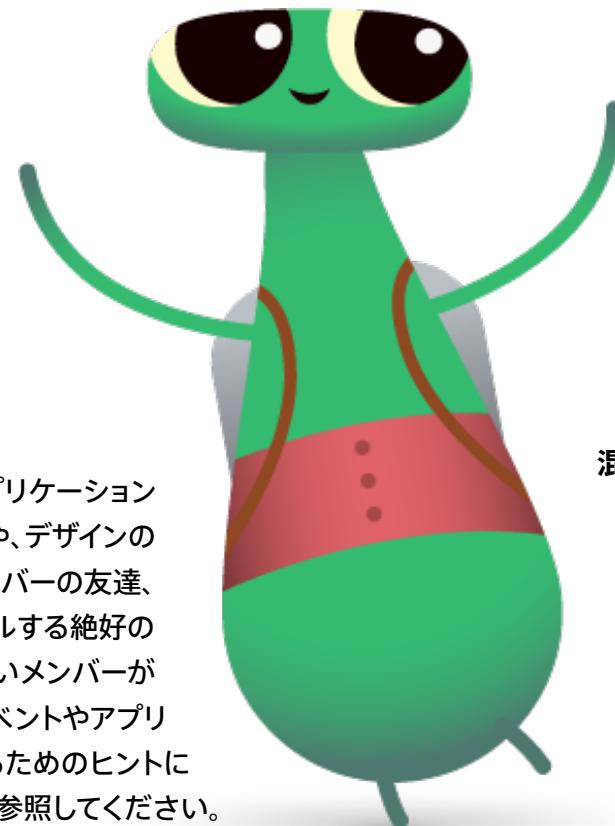


Swift Coding ClubのTシャツ



**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。**地域のイベントやアプリケーションコンテストは、クラブの活動内容や、デザインのアイデア、プログラミングスキルを、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。地域のイベントやアプリケーション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、人のためになる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。みんなが関心をもてるプロジェクトに、協力し合って取り組める方法を考えましょう。

**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 Playgroundsの使い方

クラブの教材はSwift Playgroundsを中心に作られています。Swift Playgroundsには、内蔵されているレッスンのライブラリに加え、業界をリードするデベロッパや出版者が作成した追加の課題も用意されています。まずはSwift Playgroundsのコンテンツを確認し、アプリケーションの機能に慣れるようにしましょう。



# Swift Playgroundsの機能



ページメニュー  
ページの見出しをタップすると、プレイグラウンドのすべてのページを確認できます。特定のページをタップするか、矢印を使ってページを移動します。

スピードを調整する  
コードの実行スピードを速くしたり遅くしたりできます。

実行中のコードをハイライトする  
「コードをステップ実行」を使うと、コードが1行ずつハイライト表示され、どのコードが実行されているか、理解しやすくなります。

コードライブラリ  
ツールバーをタップしてコードライブラリにアクセスし、よく使う構文をライブラリからドラッグできるので、入力操作を最小限に抑えられます。

ツール  
このメニューは、ページのリセット、写真の撮影、PDFの作成、ムービーの収録を行う時に使います。

キャラクターを選ぶ  
キャラクターをタップして別のキャラクターに変え、他の人と違う画面にしてみることもできます。

ヒント  
考え方のヒントを提示する機能です。最終的にはパズルの解き方も教えてくれますが、正解をそのままコピー＆ペーストすることはできません。先に進むには、自分でコードを入力してステップを完成させる必要があります。

# Swift Playgroundsで学習する際のヒント



**パズルの解き方を伝える。**実際にByte(バイト)の世界をライブビューで拡大したり回転したりしながら、パズルをクリアするためにやるべきことを、クラブのメンバーに理解してもらいます。2つのウインドウの境界部分を指先で押さえて左にドラッグすると、フルスクリーンで表示することもできます。

**パズルを分割する。**パズルはだんだん難しくなっていきます。クリアするために必要なステップを洗い出し、パズルを部分ごとに分けても構いません。Pagesやメモアプリケーションを使ってステップを書き出し、整理してからコードを入力するとスムーズに解くことができます。

**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。



**様々な解法で解いてみる。**どのパズルも複数の解き方があります。早めに終わったメンバーには、別の解き方を考えさせましょう。柔軟に考えたり、異なる解決方法を比較したりすることで、批判的思考力が養われます。

**ペアプログラミング。**2人のメンバーに、1台のiPadと一緒に作業させます。パズルの解き方についてお互いにアイデアを出し合い、交代しながらコードを書いてみます。

**アクセシビリティ機能を使う。**Swift Playgroundsは、iOSおよびiPadOSに内蔵された優れたアクセシビリティ機能に対応しているので、誰でもプログラミングを学べます。たとえば、色の反転やグレースケールの有効化、ズーム機能を使って可視性を高めることができます。



## 2. モジュールを選ぶ

クラブの教材は、プログラミングとクリエイティブなデザインアクティビティを織り交ぜたモジュール構成となっています。各モジュールは1時間のセッション12回分で構成されており、特定のレベルのプログラミング経験者を対象に、1つのテーマを取り上げています。「はじめよう」と「やってみよう」のセッションでは、主要な概念を学んでから、それらをSwift Playgroundsのパズルやチャレンジに応用して理解を深めます。続く「挑んでみよう」と「結びつけよう」のセッションでは、アイデアを模索したり、新しい製品を生み出すためにコードを活用する方法について考えます。プログラミングとデザインのスキルを応用し、特定のユーザーを対象としたSwift Playgroundsプロジェクトを作成またはデザインします。

各モジュールのファシリテータガイドは、本キットの後半に収録されています。すぐ確認したい場合は、以下のリンクから移動してください。

プロジェクトを作成する

### プロジェクトを作成する

iPadを使い、Swift Playgroundsの「コードを学ぼう1」と「コードを学ぼう2」でプログラミングの基本を習得します。学んだスキルを応用し、タッチ操作に反応するプレイグラウンドプロジェクトをデザインして作成します。[モジュールを表示する >](#)

アプリケーションをデザインする

### アプリケーションをデザインする

地域の問題を解決できるアプリケーションをメンバー同士で協力し、デザインします。デザインプロセスに沿って、ブレーンストーミングを行って独自のアプリケーションを計画し、プロトタイプを作成して評価する方法を学びます。[モジュールを表示する >](#)

クイズを作る

### クイズを作る

「プロジェクトを作成する」で身につけたスキルを活用して、Swift Playgroundsの「コードを学ぼう1」と「コードを学ぼう2」の難易度の高いパズルに挑戦します。ユーザー情報を要求してそれに応答するプレイグラウンドプロジェクトを作成します。[モジュールを表示する >](#)

Spheroゲームを作る

### Spheroゲームを作る

Spheroをプログラムしてクラシックなアーケードゲームをリメイクします。メンバーは一緒に取り組み、ゲームを動かしているコードを学び、そのコードを編集して独自のゲームにカスタマイズします。学んだスキルを応用し、1台または複数のSpheroロボットを使った独自のゲームをデザインします。[モジュールを表示する >](#)



### 3. 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。デザインやプログラミングのアクティビティをさらに発展させ、接続デバイスを試したり、ドローンの障害物コースやロボット救出作戦チャレンジなどを作るアクティビティを実施してみてください。

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、校外学習を実施するのもよいでしょう。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

## 地域のイベントやアプリケーションコンテスト

地域のイベントや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プログラミ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、またとない機会もあります。

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

チームごとにプロジェクトを紹介する時間を設け、簡単な質疑応答も行います。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。

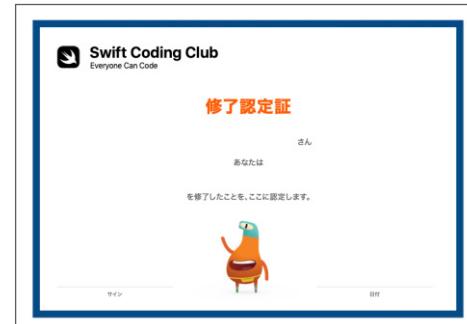
クラブのセッション中に撮った写真を楽しいスライドショーにして、最後に上映するのもおすすめです。

**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。

プログラミ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。



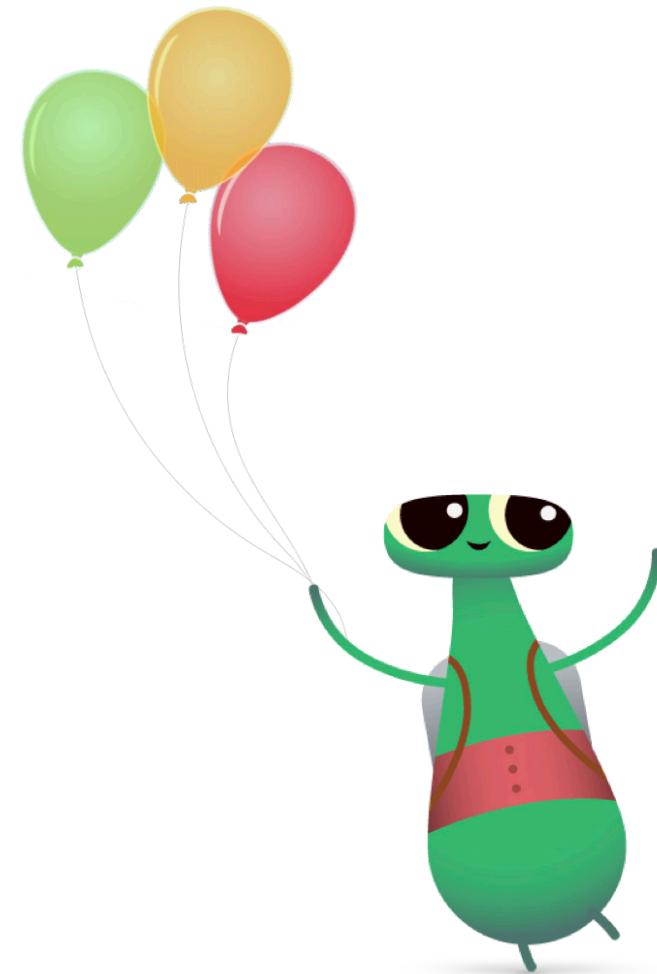
この**賞状**をダウンロードし、カスタマイズして様々な賞状を作ってください。



**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、プログラミング経験のある生徒、アプリケーション開発やデザイン業界のエキスパート、教育委員会のメンバー、自治会長、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。





# Swift Coding Club

Everyone Can Code

## 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。



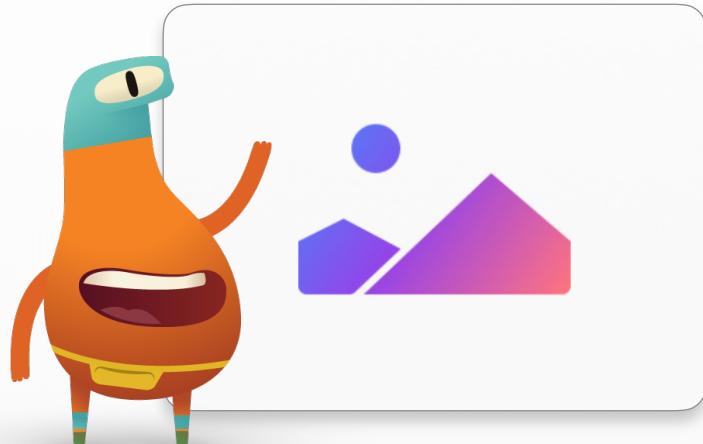
---

サイン

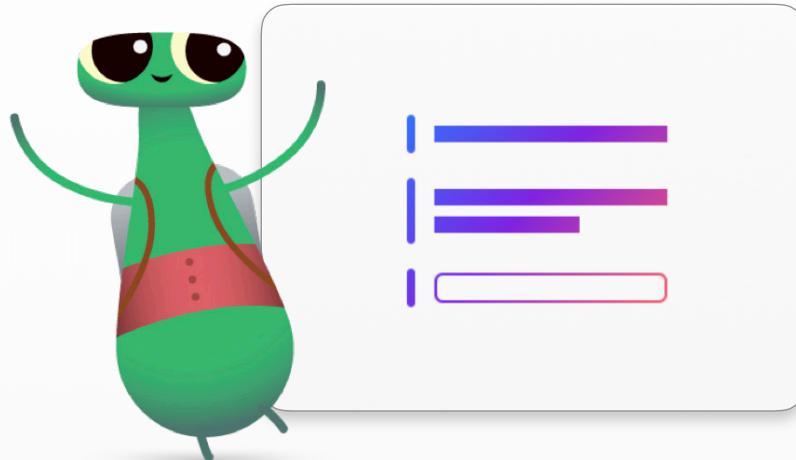
---

日付

# Swift Coding Clubモジュール



プロジェクトを作成する



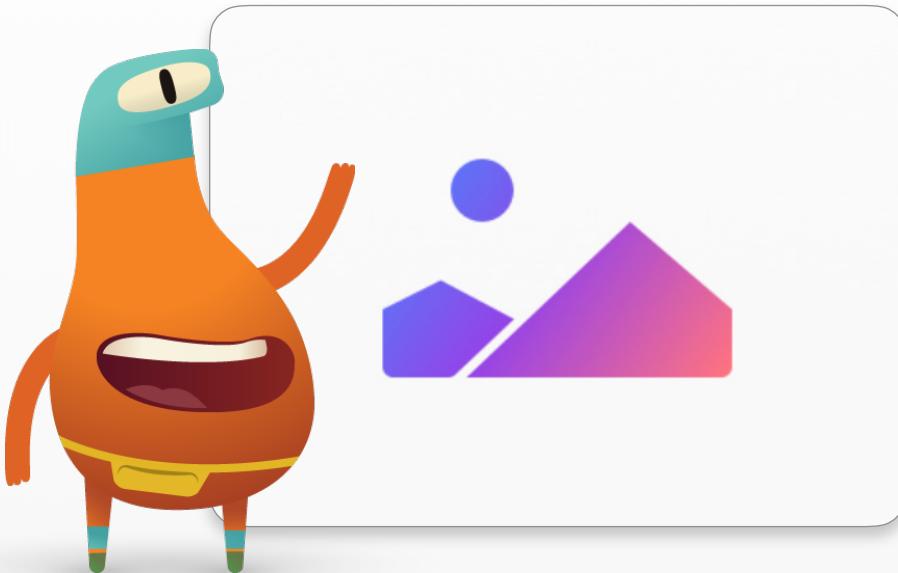
クイズを作る



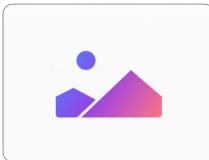
アプリケーションをデザインする



Spheroゲームを作る



# プロジェクトを作成する



# プロジェクトを作成する モジュールの概要

このセッションでは、「Everyone Can Codeパズル」ガイドに収録されている楽しいアクティビティを実施し、プログラミングの基本を習得します。メンバーはSwift Playgroundsの「コードを学ぼう1」と「コードを学ぼう2」のパズルを解いてプログラミングを練習し、学んだスキルを活用してタッチ操作に反応するプレイグラウンドプロジェクトをデザインして作成します。

「はじめよう」と「やってみよう」のセッションでは、主要な概念を学んでから、それらをSwift Playgroundsのパズルや課題に応用して理解を深めます。「挑んでみよう」と「結びつけよう」のセッションでは、アイデアを模索したり、新しい製品を生み出すためにコードを活用することについて学びます。セッションの最後には、メンバーがプロジェクトの成果を発表できる地域イベントの開催を検討してください。

各アクティビティの詳細、追加のリソース、メンバーのサポート方法、課題を発展させる方法について詳しくは、[「Everyone Can Codeパズル：教師用ガイド」](#)を参照してください。

## セッションの概要

- 「はじめよう」と「やってみよう」: 6セッション
- 「挑んでみよう」と「結びつけよう」: 6セッション
- 発表会

## リソース



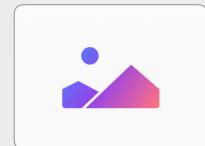
コードを学ぼう1



コードを学ぼう2



スパイラル



図形



# プロジェクトを作成する

## 1 コマンド

コマンド(コンピュータに与える特定の指示)の基本的な概念について学びます。コマンドとシーケンスを使ってコードを記述する方法も学びます。

はじめよう:「コードを学ぼう1」の「コマンド」で「はじめに」を確認する

宝探し(3ページ)

やってみよう:「コードを学ぼう1」の「コマンド」チャプタのパズルを解く(4~10ページ)

### コードを学ぼう1 コマンド

- はじめに
- コマンドを使う
- 新しいコマンドを追加する
- スイッチを切り替える



## 2 関数

関数を定義して独自のコマンドを作り、作成した関数を呼び出す方法を学びます。

はじめよう:「コードを学ぼう1」の「関数」で「はじめに」を確認する

折り紙(15ページ)

やってみよう:「コードを学ぼう1」の「関数」チャプタのパズルを解く(16~21ページ)

### コードを学ぼう1 関数

- はじめに
- 新しい挙動を作る
- 新しい関数を作る
- パターンをネストする



## 3 forループ

forループについて学び、関数とループを使ってコードをより効率よく実行する方法を確認します。

はじめよう:「コードを学ぼう1」の「forループ」で「はじめに」を確認する  
パターンメーカー(26ページ)

やってみよう:「コードを学ぼう1」の「forループ」チャプタのパズルを解く(27~31ページ)

### コードを学ぼう1 forループ

- はじめに
- ループを使う
- 4辺でループする





# プロジェクトを作成する

## 4 変数

変数を介してコンピュータが情報を保存する仕組みや、変数を使ってプログラミングする方法を学びます。

**はじめよう：**「コードを学ぼう2」の「変数」で「はじめに」を確認する

ニュースポット(36ページ)

**やってみよう：**「コードを学ぼう2」の「変数」チャプタのパズルを解く  
「スパイラル」テンプレート(37~43ページ)

### コードを学ぼう2 変数

- ・はじめに
- ・記録する



### スパイラル

- ・概要
- ・内サイクロイド
- ・外サイクロイド
- ・内トロコイド
- ・エリプシス
- ・プレイタイム



## 5 条件分岐コード

ブール論理について学び、条件分岐コードを記述する方法を確認します。

**はじめよう：**「コードを学ぼう1」の「関数」で「はじめに」を確認する

もしもゲーム(49ページ)

**やってみよう：**「コードを学ぼう1」の「関数」チャプタのパズルを解く(50~56ページ)

### コードを学ぼう1 関数

- ・はじめに
- ・スイッチを調べる
- ・else ifを使う
- ・条件分岐コードをループする
- ・関数をじょうずに使う



## 6 対象ユーザーに合わせて デザインする

ユーザーごとの視点の違いについて考え、特定の対象ユーザーに合わせて製品をデザインする方法を学びます。

**結びつけよう：**ユーザーの視点で考える(58ページ)



# プロジェクトを作成する

## 7 型と初期化

型の記述方法や、コード内で型を初期化する方法を学びます。

**はじめよう：**「コードを学ぼう2」の「型」チャプタと「初期化」チャプタで「はじめに」を確認する  
優れたデザインの特性(62ページ)

**やってみよう：**「コードを学ぼう2」の  
「型」チャプタと「初期化」チャプタのパズルを解く  
(63~66ページ)

### コードを学ぼう2 型

- はじめに
- ワープをオフにする



### 初期化

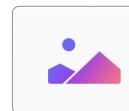
- はじめに
- エキスパートを初期化する
- 違う型のインスタンスを使う

## 8 インタラクティブな図形

Swift Playgroundsの「図形」テンプレートを詳しく確認します。この後のセッションでは、このテンプレートを使ってプロジェクトを作成していきます。「作成」「タッチ」「アニメート」のページでいろいろと試してみて、各セクションのコードで実行される内容やその仕組みについて考えます。グループで、「図形」テンプレート内で利用できるグラフィック要素と関数のリストをまとめます。

### 図形

- 作成
- タッチ
- アニメート



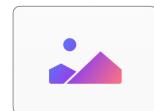
## 9 図形プロジェクトを作成する

「図形」テンプレートで、目と手をうまく協応させるためのアプリケーションプロジェクトを作成する方法を学びます。グラフィック要素と関数のリストを見直し、項目を追加します。

**挑んでみよう：**視覚運動機能を鍛えるゲームアプリケーションプロジェクトを作成する(67ページ)

### 図形

- キャンバス





# プロジェクトを作成する

## 10 プロジェクトをデザインする

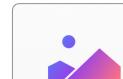
「図形」テンプレートを使って作成できるほかのプロジェクトについてブレーンストーミングします。利用できるグラフィック要素や関数を考慮して、対象ユーザーのニーズを満たすにはどうすればよいか考えます。全体でアイデアを出し合ってから、ペアになって独自のアイデアをスケッチに描き、そのプロジェクトでどのように目的を達成し、それをどのように対象ユーザーに向けてデザインするのかをまとめます。

## 11 プロジェクトを作成する

2人1組のペアになって、「図形」テンプレートの「キャンバス」ページでプロジェクトのアイデアをコードに記述します。前のセッションで作成したスケッチを参考にしながら取り組みます。

### 図形

- キャンバス

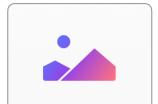


## 12 プロジェクトを評価する

プレイグラウンドプロジェクトを仲間とテストします。作品を発表する地域のイベントに備え、プロジェクトの仕組みやデザインの意図の説明を、チームで練習します。

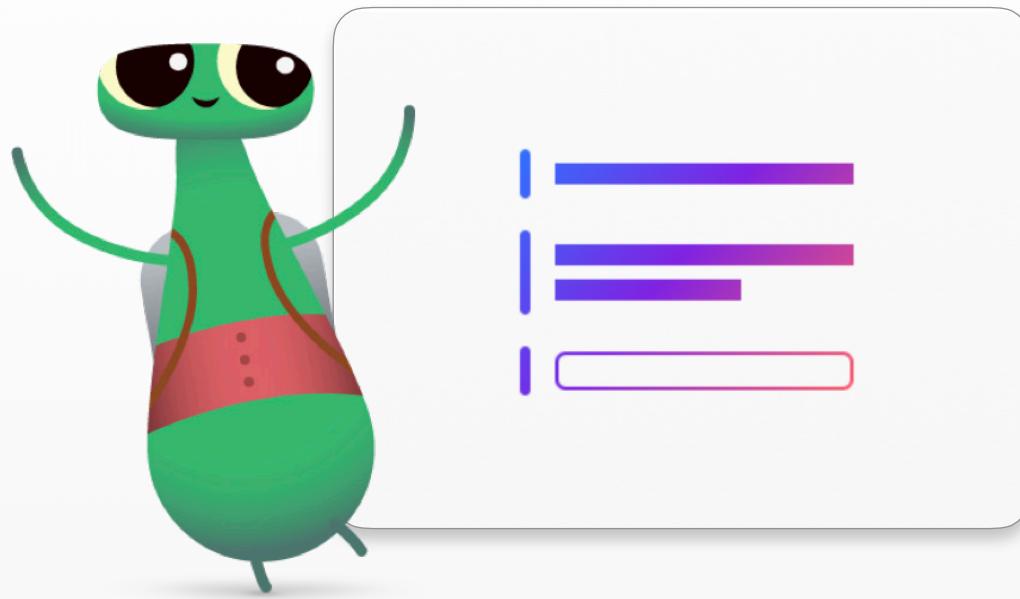
### 図形

- キャンバス



## 発表会

クラブ活動の成果を地域のイベントで讚え合いましょう。プロジェクトをデモし、デザインプロセスを説明し、参加者からフィードバックを得ることができます。



## クイズを作る



# クイズを作る モジュールの概要

このモジュールでは、「Everyone Can Codeパズル」ガイドに収録されている難易度が高いアクティビティに挑戦して、スキルを高めます。メンバーはSwift Playgroundsの「コードを学ぼう1」と「コードを学ぼう2」のパズルを解いてプログラミングの練習をします。習得した高度なスキルを応用し、ユーザー情報を要求してそれに応答するプレイグラウンドプロジェクトを作成します。このモジュールの実施には、「パズル」チャプタ1~6の内容の理解、「プロジェクトを作成する」モジュールの完了、またはこれに相当する知識が必要です。

「はじめよう」と「やってみよう」のセッションでは、主要な概念を学んでから、それらをSwift Playgroundsのパズルや課題に応用して理解を深めます。「挑んでみよう」と「結びつけよう」のセッションでは、コードを使ってアイデアを模索し、新しい成果物を作る方法を学びます。セッションが終わったら、メンバーがプロジェクトの成果をデモできる地域イベントの開催を検討してください。

各アクティビティの詳細、追加のリソース、メンバーのサポート方法、課題を発展させる方法について詳しくは、「[Everyone Can Codeパズル：教師用ガイド](#)」を参照してください。

## セッションの概要

- 「はじめよう」と「やってみよう」: 4セッション
- 「挑んでみよう」と「結びつけよう」: 8セッション
- 発表会

## リソース



コードを学ぼう1



コードを学ぼう2



じゃんけん



対話



# クイズを作る

## 1 パラメータ付き関数

関数でパラメータを指定することで、コンピューターにより詳しい情報を与える方法について学びます。

はじめよう：「コードを学ぼう2」の  
「パラメータ付き関数」で「はじめに」を確認する  
失敗しないレシピ(71ページ)

やってみよう：「コードを学ぼう2」の  
「パラメータ付き関数」チャプタのパズルを解く  
(72~75ページ)

### コードを学ぼう2 パラメータ付き関数

- はじめに
- もっと遠くまで進む



## 2 ゲームをデザインする

Swift Playgroundsの「じゃんけん」チャレンジを使って、ゲームのリメイク版をデザインします。

挑んでみよう：じゃんけんゲームを作る  
(76ページ)

### じゃんけん

- 概要
- 新しいゲームを作る
- 手を追加
- 秘密の手を追加
- 対戦相手を追加



## 3 論理演算子

論理演算子を使い、特定の条件に合わせて動作するようにコードを記述する方法を学びます。

はじめよう：「コードを学ぼう1」の「論理演算子」で  
「はじめに」を確認する  
第2回もしもゲーム(81ページ)

やってみよう：「コードを学ぼう1」の「論理演算子」  
チャプタのパズルを解く(82~85ページ)

### コードを学ぼう1 論理演算子

- はじめに
- NOT演算子を使う
- 両方正しければ合格
- 一方でも正しければ合格





# クイズを作る

## 4 クイズを作る

条件付き演算子、変数、関数、パラメータ付き関数の知識を組み合わせ、Swift Playgroundsの「対話」テンプレートで診断クイズを作ります。

挑んでみよう：診断クイズを作る(86ページ)

### 対話

- ・テキスト
- ・型



## 5 クイズプロジェクトを デザインする

「対話」テンプレートをベースにして、独自のクイズプロジェクトのアイデアを出します。クイズの目的を考え、クイズアプリケーションのデザインを模索し、対象ユーザーを検討してアイデアをスケッチします。

## 6 whileループ

whileループについて学び、指定した条件が合うまでひとたまりのコードを繰り返し実行する方法を確認します。

はじめよう：「コードを学ぼう1」の「論理演算子」で「はじめに」を確認する

公園で遊ぼう(90ページ)

やってみよう：「コードを学ぼう1」の「論理演算子」チャプタのパズルを解く(91~94ページ)

### コードを学ぼう1 論理演算子



- ・はじめに
- ・条件を決めてコードをループする
- ・whileループをじょうずに使う
- ・ループをネストする



# クイズを作る

## 7 クイズを改良する

最初に作ったクイズを更新し、whileループに異なるモードを組み込みます。このスキルは、後のセッションで独自のプロジェクトアイデアのプログラミングを行う時に役立ちます。

**挑んでみよう**：クイズを改良する(95ページ)

### 対話

- テキスト
- 型



## 8 配列とリファクタリング

このセッションでは、配列を使った新しいテクニカルスキルを学び、そのスキルを活用してコードをリファクタリングします。

**はじめよう**：「コードを学ぼう2」の「配列とリファクタリング」で「はじめに」を確認する

**評価**(99ページ)

**やってみよう**：「コードを学ぼう2」の「配列とリファクタリング」チャプタのパズルを解く(100~105ページ)

### コードを学ぼう2 配列とリファクタリング

- はじめに
- 情報をまとめて保存
- 反復処理を試す
- ブロックを積む
- 順番に並べる
- インデックスの範囲外エラーを直す



## 9 クイズに選択肢を追加する

クイズのプレイグラウンドプロジェクトに選択肢のリストを含めて更新し、選択肢リストを使って作ることができるプロジェクトについて検討を始めます。

**挑んでみよう**：診断クイズに選択肢を追加する(106ページ)

### 対話

- テキスト
- 型





# クイズを作る

## 10 新しいプロジェクトを デザインする

「対話」テンプレートを使って作成できるほかのプロジェクトについてブレーンストーミングします。全体でアイデアを出してから、それぞれ個別にアイデアを掘り下げ、目的と対象ユーザーを特定してワイヤフレームを描きます。

## 11 プロジェクトを作成する

「対話」テンプレートで独自のプロジェクトを作成します。前のセッションで作ったワイヤフレームを参考にしてください。

### 対話

- ・テキスト
- ・型



## 12 プロジェクトを評価する

プレイグラウンドプロジェクトを仲間とテストします。作品を発表する地域のイベントに備え、プロジェクトの仕組みやデザインの意図の説明を、チームで練習します。

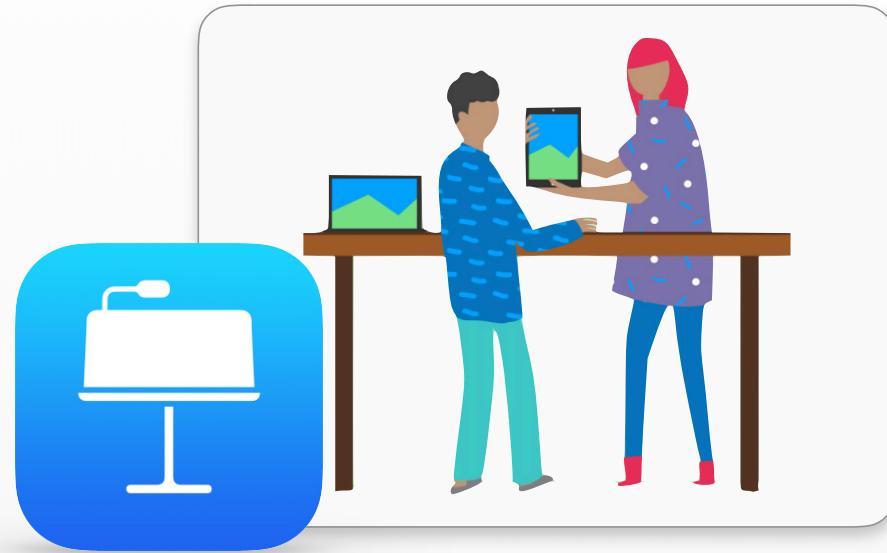
### 対話

- ・テキスト
- ・型



## 発表会

クラブ活動の成果を地域のイベントで讚え合いましょう。プロジェクトをデモし、デザインプロセスを説明し、参加者からフィードバックを得ることができます。



## アプリケーションをデザインする



# アプリケーションをデザインするモジュールの概要

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。デザインプロセスに従ってアイデアを出し合い、アプリケーションを計画し、Keynoteで実際に動くプロトタイプを作成してから、アプリケーションを評価します。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ます。

デザインプロセスの内容は「[アプリケーションデザインジャーナル](#)」で紹介されています。デザインサイクルを進めていく上で、メンバーはこのジャーナルを使ってアイデアを忘れないように書きとめておきます。プロセスを記録しておけば、アプリケーションプロジェクトの内容を繰り返し改善する際役立ちます。また、今後新規でプロジェクトを立ち上げる時のテンプレートとしても参考できます。

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讚えましょう。イベントを計画するためのヒントやリソースについては、「[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。

## 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
- コンテスト

## リソース



アプリケーションジャーナル



# アプリケーションをデザインする

## 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模索し、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ユーザーインターフェイスをデザインし、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





# アプリケーションをデザインする

## 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。



# Spheroゲームを作る



# Spheroゲームを作る モジュールの概要

このモジュールでは、メンバーはSwift Playgroundsを使ってSpheroをプログラムし、クラシックなアーケードゲームをリメイクします。メンバーにペアになってもらい、ペアごとに最低1台はSpheroを支給する必要があります。

メンバーはSpheroによって収集されるデータを調べ、それらを使って双方向型のゲームを作成する方法を学びます。ゲームの作成に必要なコードと一緒に考え、理解できたらコードを編集して動きを独自にカスタマイズしていきます。

次に、学んだ内容を応用し、1台または複数のSpheroロボットを使って独自のゲームをデザインします。メンバーには地域のイベントでゲームを披露してもらいましょう。このイベントには地域の人々を招待し、ゲームを見せたり試してもらうほか、デザインやプログラミングの工夫した点について発表する時間も設けます。

## セッションの概要

- Sphero Pong : 3セッション
- Sphero Bop It : 2セッション
- Spheroパックマン : 2セッション
- ゲームをデザインする : 5セッション
- 発表会

## リソース



Sphero Mini Robot  
(メンバーのペアに1台ずつ)



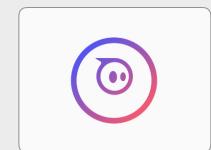
Sphero Arcade 1



Sphero Arcade 2



Sphero Arcade 3



Spheroテンプレート



# Spheroゲームを作る

## 1 Sphero Pong

Swift Playgroundsの「Sphero Arcade 1」で学習します。Spheroの動かし方を学んでから「元祖Pong」ページを開き、2人1組のペアでゲームをプレイします。このゲームを作るために必要なコードを考え、アイデアをスケッチに描いてから、擬似コードをスケッチに加えて注釈を付けていきます。

### Sphero Arcade 1

- ・はじめに
- ・ロール
- ・狙いをつける
- ・方向
- ・衝突
- ・元祖Pong



## 2~3 Sphero Pong

少人数のグループで、足をラケットにして遊ぶSphero Pongの「ライブ」ゲームを作ります。セッション3の最後に、Sphero Pongゲームの作成に必要だと思ったコードを見直し、ほかに必要なものがないか話し合います。

### Sphero Arcade 1

- ・現実世界のセットアップ
- ・跳ね返る角度
- ・行ったり来たり
- ・スコアの記録
- ・ゲームの勝敗
- ・ゲームをプレイ



## 4~5 Sphero Bop It

2人1組のペアになって、SpheroでBop Itゲームをリメイクします。「Sphero Arcade 2」で、各ジェスチャーをプログラムする方法を学び、ゲームの難易度を上げます。コードを編集して独自の動きを作り、プレイグラウンドのビジュアルインターフェイスをSpheroにリンクさせるには、ほかにどのようなコードが必要か調べます。

### Sphero Arcade 2

- ・はじめに
- ・タップ
- ・トス
- ・スピン
- ・シェイク
- ・ゲームのランダム化
- ・難易度設定
- ・ゲームをプレイ





# Spheroゲームを作る

## 6~7 Spheroパックマン

2人1組のペアになって、Spheroで「パックマン」アーケードゲームをリメイクします。「Sphero Arcade 3」で、Spheroをジョイティックとして使えるようにプログラムし、ゲームのスコアを管理して、モンスターを作成する方法を学びます。ゲームを編集して難易度を上げ、ビジュアルインターフェイスのすべての要素を実現するには、ほかにどのようなコードが必要か調べます。

### Sphero Arcade 3

- ・はじめに
- ・シンプルな操作
- ・スコアリング
- ・パワーエサ
- ・基本モンスター
- ・上級モンスター
- ・ゲームをプレイ

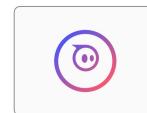


## 8~9 迷路をナビゲートする

「Spheroテンプレート」プレイグラウンドで、Spheroが迷路を抜けることができるようになります。迷路をスケッチに描き、マスキングテープで迷路を作ります。まずは簡単な迷路から始めましょう。テンプレートの「ドライブ」ページを使ってSpheroの方向を指定し、Spheroがテンプレートページの迷路を抜けることができるようになります。少人数のグループで取り組むか、簡単な迷路を作ってSpheroロボットに競争させ、誰が一番早く正確に迷路を抜けることができるか競い合ってもよいでしょう。

### Spheroテンプレート

- ・テンプレート
- ・ドライブ

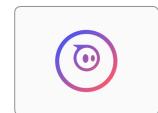


## 10~12 ゲームをデザインする

Spheroを使った独自のゲームデザインについてブレーンストーミングします。アーケードゲームの現実世界バージョンや、障害物コースチャレンジ、複数のSpheroを使ったゲームなどを考えてもらいます。ゲームをスケッチに描いて計画してから、「Spheroテンプレート」を使ってプレイグラウンドプロジェクトを作成します。ゲームを様々な要素に分解して考えること、コードにコメントを含めて自分の考えを共有することを忘れないでください。

### Spheroテンプレート

- ・テンプレート
- ・ドライブ



## 発表会

クラブ活動の成果を地域のイベントで讚え合いましょう。プロジェクトをデモし、デザインプロセスを説明し、参加者からフィードバックを得ることができます。



© 2019 Apple Inc. All rights reserved. Apple、Appleのロゴ、iPad、iPad Air、iPad mini、iPad Pro、Keynote、Mac、Pages、Xcodeは、米国および他の国々で登録されたApple Inc.の商標です。Swift、Swiftのロゴ、Swift PlaygroundsはApple Inc.の商標です。iOSは米国および他の国々におけるCiscoの商標または登録商標であり、ライセンスにもとづき使用されています。このキットに記載されたその他の製品名および会社名は、各社の商標である場合があります。2019年11月