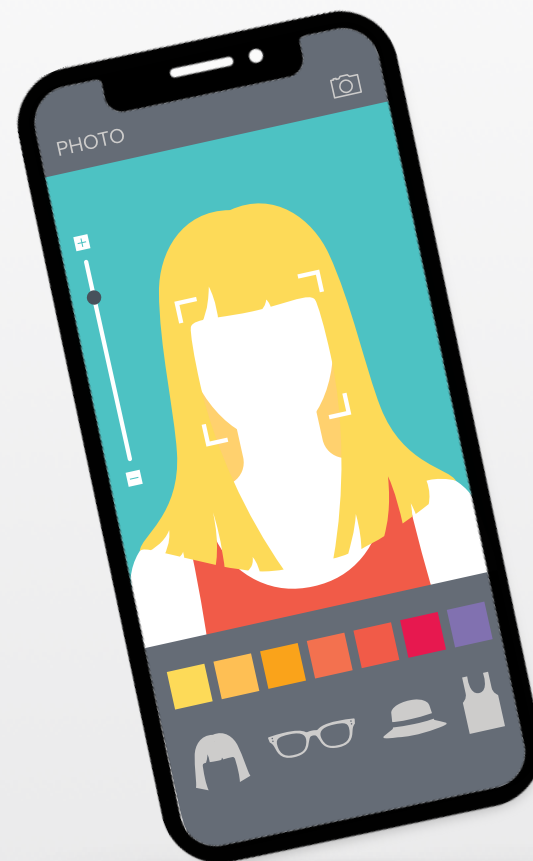




# アプリケーションコンテスト開催ガイド



# アプリケーションコンテスト アイデアを讀えましょう

アプリケーションのデザインは、自分たちが関心のある問題をチームで協力して解決するのに最適です。アプリケーションコンテストを対面またはリモート形式のイベントとして開催することで、アイデアを発表して、問題の解決策を友人や家族、地域社会と共有し、その創造力を讀める機会をプログラミングを学ぶ参加者に提供することができます。

参加者は、アプリケーションデザインのプロセスと一緒に取り組んだ後、発表用のスライドを作成し、コンテストで審査員にアプリケーションの魅力をアピールします。コンテストの締めくくりには、入賞者を発表し、参加者全員を讀えましょう。

このガイドは、Appleのプログラミング教材やリソースと併用することで、アプリケーションコンテストの計画や準備をスムーズに始められるように設計されています。お使いの環境に合うコンテストの形式や、審査員を招待して準備する際のヒントを紹介しているほか、ダウンロード可能なルーブリックや修了認定証なども用意されています。



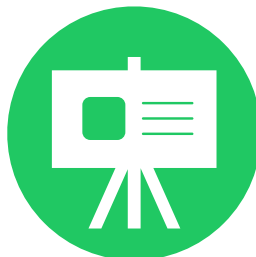
# このガイドの内容

アプリケーションコンテストの企画から開催までの、それぞれの段階で活用できるリソースとカスタマイズ可能な素材を用意しました。インスピレーションあふれるコンテストを簡単に開催することができます。



## 準備する

- ・ 3分のプレゼンテーション作成



## 計画する

- ・ コンテストの形式
- ・ コンテストの審査
- ・ 案内と宣伝
- ・ 表彰



## 共有する

- ・ コンテスト内容の投稿
- ・ 配慮すべき点



# 準備する

コンテストの参加者は、アプリケーションデザインジャーナルに沿って取り組み、発表用のプレゼンテーションを作成します。個人または少人数のチームで作業を進め、その過程で批判的思考や創造的思考力を身につけることができます。

Keynoteを使用してアプリケーションデザインをはじめる

[Everyone Can Codeの「アプリケーションデザインジャーナル」をダウンロード >](#)

さらに高度な演習やコーディングの課題に取り組む

[Develop in Swiftの「アプリケーションデザインワークブック」をダウンロード >](#)

## 3分のプレゼンテーション作成

参加者はコンテストで、アプリケーションのアイデアを3分で紹介するプレゼンテーションを行います。プレゼンテーションには以下の内容を含めます。

- ・ アプリケーションで解決しようとしている問題
- ・ アプリケーションの対象ユーザーと期待する効果
- ・ アプリケーションのデザインプロセスの説明
- ・ アプリケーションの使い方(プロトタイプのコモを含む)
- ・ プロトタイプを使ったユーザーインターフェイス、ユーザー体験、プログラミングの概念の紹介

プレゼンテーションの準備をする上でヒントになるように、評価用[ルーブリック](#)を参加者に共有しましょう。

### プレゼンテーションを行う参加者に、以下のヒントを共有します。

- ・ 個性を前面に出しましょう。
- ・ 本番に備え、プレゼンテーションの練習をしましょう。
- ・ プロトタイプのコモができるように、iPhone、iPad、Macを使って画面収録を準備しておきましょう。
- ・ Keynoteにナレーションを追加したり、ClipsやiMovieを使って、プレゼンテーションのためのビデオを作成しましょう。
- ・ オンラインでプレゼンテーションを行う場合は、チャット画面に質問が届いていないか、チームの誰かが常に確認しましょう。
- ・ Keynoteの見栄えをよくするため、ツールバー、フォーマットパネル、スライドナビゲータを隠しましょう。  
⌘Pを押すと、プレゼンテーションが開始します。

[Learning Center](#)では、教育者向けに、画面収録、Keynoteへの音声追加などに関するヒントを紹介しています。



# 計画する

## コンテストの形式

コンテストはどんな規模でも行え、対面とリモート、どちらでもかまいません。すでにあるリソースと支援してくれそうな人材を踏まえて検討しましょう。できるだけ早い段階でイベントを開催する方法や場所を決めるようにするとよいでしょう。教室、図書館、コミュニティセンターを会場にしたり、ビデオ会議システムを使ってオンラインで開催することもできます。参加者がアプリケーションをアピールできる場所ならどこでもかまいません。

ここでは、コンテストの形式の例として、アプリケーション展、ステージ発表、リモート形式のコンテストの3つをご紹介します。これらの例を参考に、参加者や地域にとって理想的な発表会のアイデアを発展させていきましょう。

### アプリケーション展

アプリケーション展では、科学展のように来場者や参加者が自分のペースで会場を回りながら、各チームの展示ブースを訪れ、プロトタイプを見ながら説明を聞くことができます。

### アプリケーション展形式の コンテストスケジュール例

- ・ 準備
- ・ 開会のあいさつ
- ・ ラウンド1
- ・ ラウンド2
- ・ 採点と評価
- ・ 表彰と修了証授与
- ・ 閉会のあいさつ
- ・ 集合写真撮影





## ステージ発表

ステージ発表の場合、チームは観客がいる中で、アプリケーションのアイデアを審査員に向けて順番に発表します。よりフォーマルな雰囲気になるため、パブリックスピーキングのよい練習機会になります。

それぞれのプレゼンテーションの後にチームに質問できるように、審査員にはステージ上またはステージの近くに座ってもらいましょう。

### ステージ発表形式の コンテストスケジュール例

- ・準備
- ・開会のあいさつ
- ・プレゼンテーション1
- ・審査員からのフィードバックと質問
- ・プレゼンテーション2
- ・審査員からのフィードバックと質問
- ・プレゼンテーション3(以下繰り返し)
- ・採点と評価
- ・表彰と修了証授与
- ・閉会のあいさつ
- ・集合写真撮影



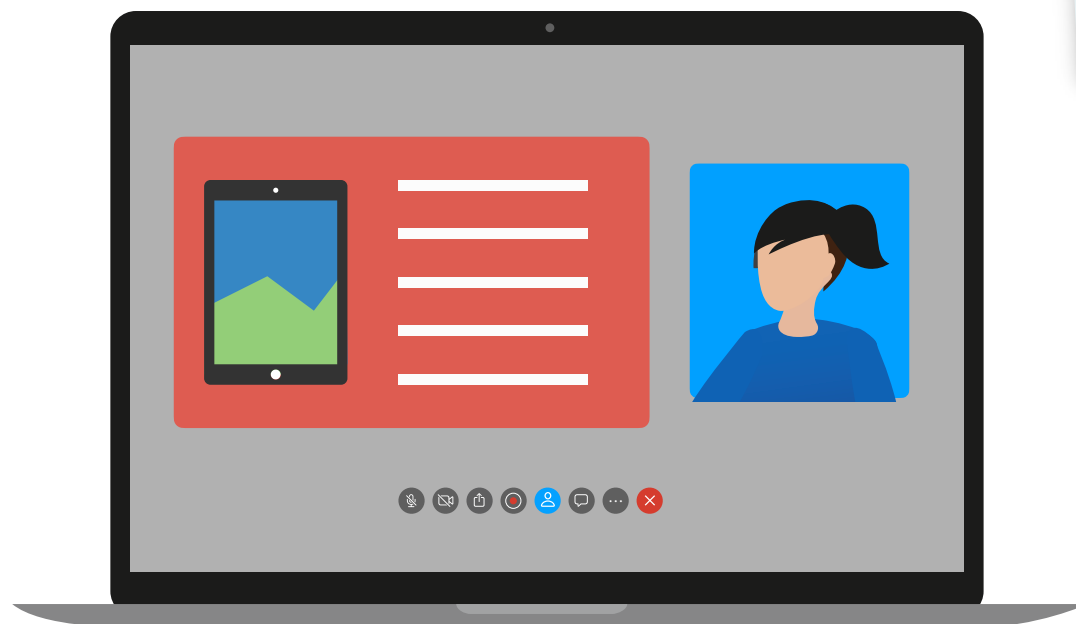


## リモート形式のコンテスト

リモート形式のコンテストでは、各チームがビデオ会議ツールを使ってオンラインでプレゼンテーションを実施します。どのような学習環境にいる場合でも、楽しみながら参加できる方法となるでしょう。

参加者がイベントの形式を理解し、プレゼンテーションを練習できるように、事前にリハーサルを行うことを検討してください。コンテスト当日は、開場時に音楽をかけたり司会者が参加者を歓迎したりして、楽しいイベントの雰囲気を演出しましょう。

審査員がビデオ会議に参加できない場合は、画面収録を共有してもかまいません。審査員には、参加者へのフィードバックをビデオまたは文章で提出してもらいましょう。



### リモート形式の コンテストスケジュール例

- ・ 発表者の事前準備
- ・ 開会のあいさつ
- ・ プレゼンテーション1
- ・ 審査員からのフィードバックと質問
- ・ プレゼンテーション2
- ・ 審査員からのフィードバックと質問
- ・ プレゼンテーション3(以下繰り返し)
- ・ 休憩(採点と評価)
- ・ 表彰
- ・ 閉会のあいさつ

### リモート形式のコンテスト開催時のヒント

- ・ 共同開催者を指名し、観客からのチャットやテキストメッセージの質問に回答できるようにしましょう。
- ・ 発表者以外の参加者をすべてミュートにして、雑音を抑えましょう。
- ・ 審査員がプレゼンテーションについて協議できるように、別回線の通話またはビデオ会議を用意しましょう。
- ・ スクリーンショットを撮影してコンテストのコラージュ写真を作り、参加者に共有しましょう。
- ・ イベント終了後は、参加者に参加証明書をメールで送りましょう。



## コンテストの審査

審査員を招くことで、参加者は自分のアイデアについてより多角的なフィードバックを得ることができます。審査員はプログラミングの専門家である必要はありません。地域の事業者や自治会長などに依頼するのもよいでしょう。リモート形式でアプリケーションコンテストを開催すれば、遠方にいる審査員も招待しやすくなります。積極的に声をかけるようにしましょう。

### 審査のヒント

- ・審査員と一緒に、評価用**ループバック**を確認します。どのような賞を計画しているか審査員に伝えましょう。
- ・審査員から参加者に質問とフィードバックをしてもらいます。
- ・審査員が集まって評価を協議するための、静かな場所を用意します。
- ・リモート形式のコンテストでは、審査員に別回線の通話またはビデオ会議で集まってもらうとよいでしょう。

### 審査員依頼のメール例

【ご協力のお願い】 アプリケーションコンテストへのご参加について  
[名前]様

[組織の名称]では、[日付][時間]時から[時間]時まで、アプリケーションコンテストを開催する予定です。つきましては、[名前]様に審査員としてご協力いただきたく存じます。参加をお願いすることは可能でしょうか。[名前]様から直接アドバイスをいただけると、プログラミングを学ぶ参加者にとって、この上なく貴重な体験となるかと思えます。

審査員の皆様には、参加チームが行う短いプレゼンテーションをお聞きいただき、それに対して質問したり、フィードバックをした後に、こちらで用意する採点表を使って各プレゼンテーションを評価していただきたいと思います。締めくくりとして、今まで学んできたことの集大成を披露してもらい、参加者を讃える時間も設けています。

ご検討いただき、出席の可否をお返事をいただけますと幸いです。地域の若者に対する皆様のご支援に深く感謝いたします。また、ご不明な点があればご遠慮なくお知らせください。

よろしくお願い申し上げます。

[学校または組織の名称]

[役職]

[名前]







## 案内と宣伝

コンテストへの期待感を高めて、地域の人たちの参加を促すためのアイデアをいくつかご紹介します。

- ・ 家族や友人、自治会長といった特別ゲストに案内状を送ります。
- ・ 発表者に友人や家族を招待するよう促します。
- ・ 所属組織のウェブサイト、ソーシャルメディア、ニュースレターなどでコンテストを宣伝します。

### コンテストの案内状サンプル

素晴らしいアイデアの数々をご覧ください。

[組織名]では第1回アプリケーションコンテストを開催します。参加者が審査員にプレゼンテーションを行い、今まで学んだことの集大成を披露します。関心のある問題を解決するアプリケーションのアイデアを発表し、競う場にぜひ応援にいらしてください！

概要

[日付]

[時間]

[会場名]

[会場の住所]

[問い合わせ先]

[申し込み用サイトのリンク]



## 表彰

楽しく競い合うことで学習意欲が高まります。以下のような賞を設けて、参加者のアプリケーションデザインが優れていた点を表彰することを検討してもよいでしょう。

- ・最優秀アイデア賞
- ・最優秀デザイン賞
- ・最優秀プレゼン賞

参加者全員に修了認定証を授与します。また、特別賞を追加し、観客に投票してもらうこともできます。こちらの修了認定証のテンプレートを[ダウンロード](#)し、カスタマイズしていろいろな賞状を作ってみてください。



参加者には、発表会の前や当日にお揃いのイベントTシャツを配るのもよいでしょう。Tシャツのデザインテンプレートは[こちら](#)からダウンロードできます。

# 共有する



## コンテスト内容の投稿

アプリケーションコンテストを開催することで、みなさんは未来のイノベーターを育むコミュニティの一員となります。ぜひTwitterで@AppleEDUをフォローしてください。また、ツイートに「#EveryoneCanCode」や「#DevelopInSwift」のハッシュタグを付けていただくと、私たちともコンテストの様子を共有することができます。

## 配慮すべき点

- ・ ソーシャルメディアに関する所属組織のガイドラインと規則を必ず確認します。
- ・ 参加者の作品を共有する前に、保護者の同意を必ず得るようにします。
- ・ 自分たちの知的財産を保護し、他者が有する著作権を尊重するよう、参加者に指導します。



評価用ルーブリック

ダウンロード >

チーム名： \_\_\_\_\_

	もう一步 (1点)	できました (2点)	よくできました (3点)	非常によくできました (4点)	得点
プレゼンテーションの内容	アプリケーションの目的や対象ユーザーなど、基本的な情報を共有している。	アプリケーションの目的とデザインの方 向性、それがどうユーザーのニーズと結びついているかについて、明確に説明している。	解決しようとしている問題、市場のニーズ、対象ユーザー、ユーザーのデザインニーズについて、明確かつ有効的に提示している。	アプリケーションがどのようにユーザーのニーズを満たし、生活を変えるのかについて、エビデンスを示しながら説得力を持ったプレゼンテーションをしている。	
プレゼンテーション技術	重要項目を説明している。	自信と熱意を持ってプレゼンテーションを行っている。	聞き手を引きつける様々な方法を使って魅力的なプレゼンテーションを行っている。	独創的で、印象的なストーリーで構成されている。視覚要素をうまく使い、聞き手を引きつけながら発表できている。チームメンバー間の連携もスムーズである。	
ユーザーインターフェイス	UIデザインによりアプリケーションの機能が高められており、すべての画面でテーマが一貫している。	UIデザインには、iOSでよく使われるインターフェイス要素、アイコン、テキストスタイルなどが採用されており、使いやすく機能性に優れている。プロトタイプには、アプリケーションの動作がわかるインタラクティブな要素が含まれている。	美しく、シンプルでストレスなく使えるUIで、色、レイアウト、読みやすさが細部まで考慮されている。ユーザーが困らないよう、ナビゲーションが工夫されている。	デザイン性が高く、もっと使ってみたくなる仕掛けがある。アニメーションや色、レイアウトを工夫し、シームレスで楽しい体験が提供できている。	
ユーザー体験	アプリケーションの目的や、アプリケーションを操作して目的を達成する方法が明確に表現されている。	一貫性があり、標準的なナビゲーションが実装されている。直感的に画面移動ができる。	ユーザーのニーズに柔軟に対応できる構成になっている。アクセシビリティやプライバシー、セキュリティを考慮した機能が含まれている。	革新的で驚きのある体験が提供されている。ほかのチームや既存のアプリケーションには見られないスタイルや個性が表現されている。	
コーディングの概念	アプリケーションの機能と実装するコードに関連性が見られる。	データ型、条件ロジック、タッチイベントなどの基本的なコーディングの概念と、アプリケーションとの関係について説明している。	必要なコーディング作業の具体的な説明があり、そのコードによってどのような機能が実装されるのかが、わかりやすく提示されている。	アプリケーションのアーキテクチャ、データ構造、アルゴリズム、および機能についての説明があり、なぜそのような選択をしたのかについての過程が説明されている。	
技術的観点 (任意)	Swiftコードが特定の条件で動く。Xcodeを使ったプロトタイプを審査する場合および、評価者がSwiftとiOSの開発についての知識を持ち合わせている場合に使用。	あらゆるケースでエラーなく実行できる。基本的なSwiftコードであり、ある程度抽象化されている。	Swiftの命名規則が使用されており、整理されて読みやすく、高いレベルで抽象化されている。iOSのガイドラインに準拠している。	コードはコメントを使って十分に説明されている。Swiftの機能が効果的に使われている。MVC (Model View Controller) などの標準的なデザインパターンが採用されている。	
コメント：					0 合計点



アプリケーションコンテスト

# 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

