「総合的な学習の時間」に プログラミングを取り入れる

Everyone Can Codeを使った問題解決

プログラミングの基礎を学んで、地域の課題を解決するアプリケーションをデザインしよう

2020年度からの小学校プログラミング教育の実施をふまえ、文部科学省では総合的な学習の時間を活用し、探求活動の一環としてプログラミングを授業に取り入れることができるよう、35時間の指導案を公開しています。教師のみなさんは、指導計画例にそってAppleが開発したEveryone Can Codeのレッスンに取り組み、iPadとSwift Playgroundsアプリケーションを使ってプログラミング的思考を身につけながら、地域の問題解決に取り組めます。

学習の目的

身のまわりにある問題や身近な地域の課題に気づき、チームワークを発揮しながらクリエイティブな方法で課題を解決するためにプログラミングの基礎を学び、アプリケーションをデザインして発表します。課題解決を通して、生徒たちはより良い地域にするために自分たちでできることを自発的に考えたり行動したりするようになります。

プログラミング学習の分類

A分類（学習指導要領に例示されている単元内での実施）の総合的な学習の時間（情報に関する探求的な学習）にて実施することを想定しています。総合的な学習の時間だけでは35時間を確保できない場合は、社会科や図工のB分類（学習指導要領に例示されていはいないが、学習指導要領に示される各教科等の内容を指導する中で実施するもの）、またはC分類（教育課程内で各教科等とは別に実施するもの）を組み合わせて実施すると良いでしょう。

対象：小学校高学年以上

低学年、中学年での生活科や総合的な学習の時間で行ってきた地域の学習を踏まえて、自分たちが暮らす地域の課題に気づき、プログラミングの力で解決する方法を探ります。プログラミング初心者であれば、中学校や高等学校での総合や情報の時間でもお使いいただけます。

関連する学習内容：小学校低学年

身近な人々、社会及び自然と関わる活動についての学習。中学校の技術（情報の技術）や社会（公民的分野）を踏まえた上での活動にすることも可能。

学習の流れ

指導案は3つのステージで構成され、それぞれのステージは指導要領に沿って課題の設定、情報の収集、整理・分析、まとめ・発表の4ステップに分かれています。1と2の順番を入れ替えたり、確保できる時間数が35時間を下回っても問題ありません。

1. 身近な問題に気づく（6時間）

地域の問題について調べたり、家族や専門家にインタビューし、気づいたことと、その解決案をKeynoteなどのアプリケーションを使ってまとめ、発表します。

2. プログラミングの基礎を学ぶ（13時間）

Swift Playgroundsアプリケーションを使ってプログラミングの基礎を学び、問題解決の思考を養います。

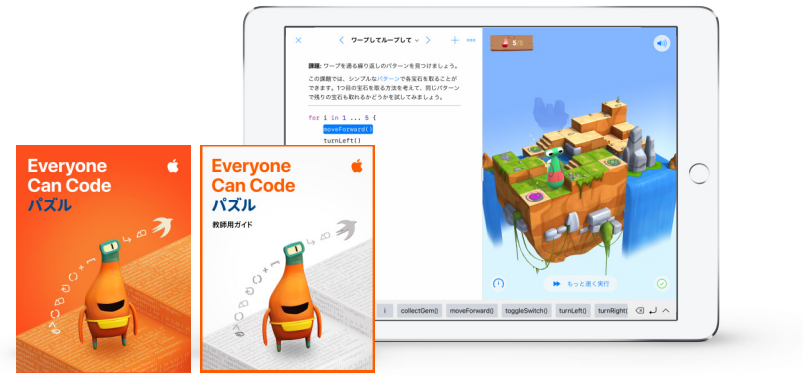
3. アプリケーションをデザインする（16時間）

問題を解決するために有効なアプリケーションのアイデアを考え、Keynoteを使ってプロトタイプ（試作品）をデザインし、発表します。

Everyone Can Code

誰でもプログラミングを学び、教えることができるようAppleが開発した、プログラミング教材です。パズルを解いてプログラミング的思考を養うところから、本格的なアプリケーションの開発まで、段階的、体系的に楽しく学び続けることができるよう設計されています。それぞれのレベルに教師用ガイドが準備されており、この指導案のステージ2は、「Everyone Can Codeパズル」、ステージ3は、アプリケーションデザインジャーナルの内容に準拠しています。

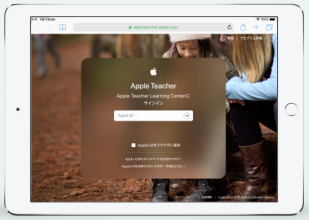
Everyone Can Codeについて詳しく >



ステージ1：身近な問題に気づく

ヒント

Apple Teacher Learning Centerの「iPadのための新しいスキル」を参考にすると、iPadの基本や創造的な学びのヒントを得ることができます。



1 課題の設定
(1時間)

日常生活に目を向けてみます。普段の生活の中で、自分自身や家族、友達が困っていること、抱えている問題について話し合ってみましょう。同じ視点で、地域の問題にも注目します。暮らしやすさにつながっている要素と解決したい問題について、自分が知っていることや調べてみたいことを書き出します。例えば、通学路道路に潜む段差など、自分の生活にも関わる安全性の問題や、防災、教育、環境(ゴミ、美化)、自然保護、高齢化、福祉・健康、観光など、身のまわりのことから考えていきます。

2 情報の収集
(2時間)

本やインターネットで問題について調べたり、家族に取材してみます。どのような質問をすれば問題点が浮き彫りになるか、必要な情報は何かの見通しを立て、探求しましょう。ゲストティーチャーを招き、地域の取り組みについての話をしてもらったり、地域の方にも取材してみます。たとえば以下の点について質問してみます。

- どのようにして問題を把握しているか?
- 問題の解決にどのようなアプローチが用いられているか?
- 地域の人たちに問題を知らせるために、どのような告知、PR活動が行われているか?

3 整理・分析
(2時間)

集めた情報を整理して、自分たちが解決できそうな問題を選びます。

- 問題を視覚的に整理してみましょう。マインドマップやウェビングを使うと便利です。
- それぞれの問題について、解決策のアイデアをたくさん出してみましょう。
- 問題解決にあたり、いま自分たちができること、できないことを判断します。

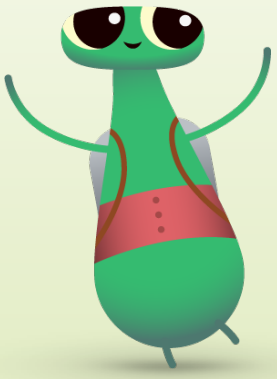
4 まとめ・発表
(1時間)

整理できた問題をKeynote、Pages、Clips、iMovieなどを使ってまとめ、自分たちの考える問題を発表します。地域には様々な問題があることを認識し、ジブンゴトとしてとらえることを目指しましょう。

ステージ2：プログラミングの基礎を学ぶ

ヒント

「Everyone Can Codeパズル」のチャプタ1から3までを目標に実践してみましょう。ステージ2の詳細な授業案やSwift Playgroundsの解答は、「Everyone Can Codeパズル：教師用ガイド」からダウンロードできます。



1 課題の設定
(1時間)

テクノロジーが「今」の社会にもたらしている効果について学習します。例えば、朝起きてから夜寝るまでの行動を振り返り、テクノロジーとの関わりを考えてみます。デジタル放送、信号、エレベーター、ゲーム、オンラインショッピングなど、思いついたものを挙げ、プログラミングがもたらす「これから」の可能性を身近なところから考えてみます。

- どこにいても学べ、働ける
- どこにいる誰とでもつながれる
- よいアイデアさえあれば、誰でもどこからでも簡単に形でき、世界に届けられる
- より多くの情報にアクセスし、つなぎ合わせ、今までできなかったことを実現する
- 結果、新しいスキルや仕事が生まれる

問題解決には様々な方法があることを生徒が気づき、さらに多くのテクノロジーはプログラミングの力で動いていること、その仕組み(アルゴリズム)は人間が考え、運用していることを紹介して、生徒からプログラミングへの興味を引き出します。

2 情報の収集
(8時間)

「Everyone Can Codeパズル : 教師用ガイド」に沿って、はじめよう(アンプラグドプログラミング)、やってみよう(Swift Playgroundsを使ったプログラミング)、結びつけよう(振り返り)のプロセスを体験します。挑んでみよう(発展活動)はスキップしても構いません。また、テクノロジーが身近な生活を良くする例として、AR(拡張現実)を体験することもできます。

- AR は世の中でどのように活用されているでしょうか?
- iPad内蔵のARアプリケーション「計測」などを実際に使ってみましょう
- ARテクノロジーを応用するとどのような問題が解決できそうか、アイデアを話し合ってみましょう

3 整理・分析
(2時間)

身近な生活を良くしているアプリケーションについて調べたことを整理します。

4 まとめ・発表
(2時間)

プログラミング体験を踏まえて、情報技術の進化が人の生活を豊かに、幸せにしていく未来についての自分の考えをまとめてみよう。自分が考える未来の形をイラストやエッセイでまとめよう。

ステージ3：アプリケーションをデザインする

ヒント

アプリケーションデザインジャーナルをダウンロードし、ワークシートやポートフォリオとして活用することができます。Keynoteにリンクを貼って、アプリケーションのように動作させる方法（プロトタイプ）も説明されています。

1 課題の設定
(1時間)

アプリケーションデザインジャーナルを参考に、ブレーンストーミング、計画、プロトタイプ、評価の4ステップの「デザインサイクル」を通してアイデアを形にしていきます。まずは、テクノロジーの力で地域の暮らしを良くする例を学んだことを踏まえ、ステージ1で設定した地域の問題に立ち戻って、解決方法をブレーンストーミングしてみましょう。ステージ1では「できない」と思ったことも、プログラミングという手段を使えば解決できるかもしれません。それらのアイデアをアプリケーションにすると、どんな内容になるでしょう？

2 情報の収集
(2時間)

自分たちがデザインするアプリケーションの使い手はどのような人か、考えてみます。世の中には様々な人がいます。誰にとっても使いやすいものにするには、どのような機能が必要か、その人になったつもりで考え、リスト化しましょう。Appleのウェブサイト上のアクセシビリティに関するビデオや機能の紹介を参考にしながら、iPadに内蔵されている機能を実際に使ってみます。iPadの「設定」にある「アクセシビリティ」を調べ、自分や家族にとっても便利な機能があるか、確認します。

3 整理・分析
(10時間)

デザインサイクルの計画ステップに移ります。アプリケーションの目的や対象ユーザーを考慮し、デザインを考えます。機能や画面のイメージを手書き入力できるアプリケーションや、写真を使ってフローチャートとして表現してみます。アイデアを発表し、みんなからのフィードバックを反映してデザインコンセプトを最終化させたら、アプリケーションに必要なイラスト、写真や画像、音声などを用意し、Keynoteを使ってプロトタイプとしてまとめます。

4 まとめ・発表
(3時間)

アプリケーションデザインを発表します。自分の作品を評価するため、保護者や地域の人に試してもらい、アプリケーションのアイデアや使いやすさについてコメントやアドバイスをもらいましょう。自分の作品を振り返り、問題の解決につながるものであったかどうかをあらためて考えます。改善したり、最初に決めた課題の設定やデザイン全体を変える必要があるかなどを総合的に見直します。改善案や変更案を具体化し、今後につなげましょう。また、プログラミングを学ぶことはなぜ大切なのか、どのような意味があるのか、考えて話し合います。自分や自分のまわり、世界中の様々な境遇にある人の立場からも考え、自分の言葉で表現してみます。

教師のみなさんのための資料とサポート

スキルを磨く

Today at Apple

Apple StoreではApple製品の活用方法を学ぶ、無料のToday at Appleセッションを毎日開催しています。お近くの店舗のSwift PlaygroundsやKeynoteを使ったアプリケーションのプロトタイプについて学ぶセッションのスケジュールを確認しましょう。ARやアクセシビリティについてのセッションもあります。

Today at Appleに参加 >

Apple Teacher

Apple Teacherは指導と学習にApple製品を組み込んでいる教育者のみなさんを支援し、その成果をたたえるための無料のプログラムです。アプリケーションの使い方やプログラミングの実践についてオンラインで学んでみましょう。

Apple Teacherに参加 >

授業で提示する

「Everyone Can Codeパズル」

Everyone Can Codeには生徒用ガイドと教師用ガイドがあります。生徒用ガイドを教師の皆さんのデバイスにダウンロードして、提示用教材として授業内で活用しましょう。生徒のiPadにもダウンロードしておくと、より効果的です。

「Everyone Can Codeパズル」生徒用ガイドをダウンロードする >

アクセシビリティ

テクノロジーが障害を持つ人の生活をどのように豊かにしているか、ビデオによるストーリーを交えながらアクセシビリティの機能についてWeb上で学ぶことができます。

アクセシビリティについてもっと詳しく >

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ジャーナルでは、プロトタイプの作品例を紹介しています。27ページのiPad向けプロトタイプKeynoteをダウンロードし、活動のイメージを共有しましょう。

アプリケーションデザインジャーナルをダウンロードする >
プロトタイプをダウンロードする >

使用するアプリケーション

すべてのアプリケーションは、App Storeで無料で入手できるか、iOSに搭載されています。

調べる

Safari

ブック

プログラミング

Swift Playgrounds

AR

計測

創る、まとめる

Pages

Keynote

Numbers

GarageBand

Clips

iMovie

Tayasui Sketches School

© 2019 Apple Inc. All rights reserved. Apple、Appleのロゴ、iPad、Keynote、Pages、GarageBand、iMovieは、米国および他の国々で登録されたApple Inc.の商標です。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、ライセンスに基づき使用されています。このガイドに記載されているその他の製品名および社名は、帰属する各社の商標である場合があります。2019年11月