これからつくる

iPhoneアプリ開発入門

Swiftではじめるプログラミングの第一歩

藤治仁　徳弘佑衣　小林加奈子　小林由憲 =著

SB Creative

本書に関するお問い合わせ

この度は小社書籍をご購入いただき誠にありがとうございます。小社では本書の内容に関するご質問を受け付けております。
本書を読み進めていただく際に不明な箇所がございましたらお問い合わせください。
なお、お問い合わせに関しましては以下のガイドラインを設けております。恐れ入りますが、ご質問の際には最初に下記ガイドラインをご確認ください。

ご質問の前に

小社ウェブサイトで「正誤表」をご確認ください。最新の正誤情報を下記のウェブページに掲載しております。

本書サポートページ：http://isbn.sbcr.jp/87148/

上記ページの「正誤情報」のリンクをクリックしてください。なお、正誤情報がない場合、リンクをクリックすることはできません。

ご質問の際の注意点

●ご質問はメール、または郵便など、必ず文書にてお願いいたします。お電話では承っておりません。
●ご質問は本書の記述に関することのみとさせていただいております。
　従いまして、○○ページの○○行目というように記述箇所をはっきりお書き添えください。
　記述箇所が明記されていない場合、ご質問を承れないことがございます。
●小社出版物の著作権は著者に帰属いたします。従いまして、ご質問に関する回答も基本的に著者に確認の上回答しております。
　これに伴い返信は数日ないしそれ以上かかる場合がございます。あらかじめご了承ください。

ご質問送付先

ご質問については下記のいずれかの方法をご利用ください。

ウェブページより
上記のサポートページ内にある「この商品に関する問い合わせはこちら」をクリックすると、メールフォームが開きます。
要綱に従ってご質問を記入の上、送信ボタンを押してください。

郵送
郵送の場合は下記までお願いいたします。

〒106-0032
東京都港区六本木2-4-5
SBクリエイティブ　読者サポート係

はじめに

この書籍は、「iOSアプリを作ってみたい、すべての初心者が、体験しながら学べる入門書」です。

iOSはAppleがMac OSをベースに開発した、iPhone、iPad、iPod Touch向けのモバイルOSです。

執筆陣は本書の構成段階からハンズオンセミナーを開催し、たくさんの初心者の方々の声をまとめ、各レッスンを構成しました。

レッスンごとに、ハンズオンセミナー開催後に、参加者の方々からフィードバックをいただく時間を設け、iOS開発の初心者の方々がわからないポイントはどこで、どの操作でつまずくのかを丁寧に調査しました。

また、執筆陣は「Swiftビギナーズ倶楽部」というコミュニティを継続開催しており、プログラミング自体がはじめての方からたくさんの意見を取り入れています。

そうした調査に基づき、プログラミングを通して、モノづくりの楽しさを体験していただけるように、少しずつ階段を登っていく体験を重視した構成にしました。初心者が最初の一歩を踏み出す書籍を目指しています。

本書のコンセプトは「まずは体験してみて、その経験を生かして学んでいく」です。

難しいプログラミング文法の説明は極力最小限にまとめ、多くのサンプルアプリの開発を体験してもらうことで、最短距離でアプリ開発の「勘所」がつかめるように工夫しました。

執筆陣は、iOSアプリ開発者や勉強会の主宰者、セミナーの講師、イベントでのスピーカーなど、多彩な経験者で構成されています。

これからiOSアプリを制作される方のために、思いを込めて執筆しました。

ようこそ、iOSアプリ開発の世界へ。

目　次

Contents

Lesson 4

楽器アプリを作ろう　～音の扱い方を学ぶ～

音もアプリのクオリティを高める大切な要素になります。
ここでは、iOSアプリ開発で音をどのように実装していけばよいのかを体験していきましょう！

Lesson 5

マップ検索アプリを作ろう ～UIパーツの扱いとdelegateを学ぶ～

テキストエリアにキーワードを入力すると、
該当する場所にピンが立つ、マップアプリを開発していきます。
文字を入力できるUIパーツ Text Fieldは、キーボードのリターンキーなどを
カスタマイズすることができますので、その設定方法も学んでいきます。

2日目

Lesson 1

タイマーアプリを作ろう ～画面遷移とデータの保持について学ぶ～

ここでは、画面遷移の方法を学びます。タイマー画面と設定画面の2つの画面がある、
タイマーアプリを作ります。タイマー画面で秒数のカウントダウンを行い、
設定画面でタイマーの秒数を選ぶことができます。

Lesson 2

SNS投稿ができるカメラアプリを作ろう ～前半～

カメラを利用することができる「UIImagePickerController」と、カメラロールに保存したりSNS投稿ができる「UIActivityViewController」を利用して、カメラアプリを開発していきます。画面のレイアウトを設定するAutoLayoutの機能や、UIパーツとプログラムを関連付けする方法も学んでいきましょう。

Lesson 3

SNS投稿ができるカメラアプリを作ろう ～後半～

カメラアプリには、エフェクト機能が定番機能になっていることが多いです。エフェクト機能を利用できる「Core Image」を利用して開発していきます。また、画面遷移を実現する「Segue」の応用として、画面遷移するときに情報を渡すことを学びます。

Lesson 4

お菓子検索アプリを作ろう ～Web APIとJSONの使い方を学ぶ～

ネットからお菓子の情報を「JSON」形式で取得し、iPhoneに表示してみましょう。ここでは、インターネットのデータを有効活用するために、WebAPIとJSONについて学んでいきます。そして、取得したデータをTable Viewを使って、リストで表示します。

0：この本の読み方と使い方

0-1 本書が対象とする方

・プログラムを書いたことはないけれど、iPhone/iPadアプリを作ってみたい方。
・iPhoneアプリをよく利用していて、自分でも作ってみたいと思った方。
・中高生、大学生でiPhoneアプリ開発を学んでみたい方。
・シルバー世代や中高年の方で再学習を実施したい方。
・企業で入社前研修や企業導入研修での教材を検討している方。

そんなiOSアプリを作ってみたい、すべての初心者が対象です。
アプリを作ることを「開発」するともいいます。
開発といっても「難しいことをする！」と身構える必要はありません。
プログラミングを楽しみながら、リラックスして読み進めてください。

0-2 本書でできるようになること

初心者の方もサンプルアプリを作ることにより、**動く体験と基本の知識が身につく**ようになります。

この書籍を終えるころには、他の入門書やプログラミング文法書を読む力もついていると思います。また、作りたいアプリや学習したい分野も見えてくると思いますので、ぜひ、次の書籍を購入してステップアップしていきましょう。

0-3 本書の特徴

とにかく「体験」すること、そしてあとから「理解」することに重点を置いています。

本書では、プログラミングの文法説明は最小限にして、iPhone/iPadアプリを作って動かしていくことを目的として構成しています。

プログラミング文法書のように文法を理解し、覚えるのではなく、どんどんアプリを作って体験していくことに比重を置いています。プログラミングがはじめての人でもiOSアプリが作れるという体験ができるように工夫しました。

学習が進めやすいように、学校の授業のように時限制（レッスン）で区切っています。

各レッスンごとに独立したサンプルアプリが作れるように配慮していますので、制作したいサンプルアプリがあれば、途中からでも学習できます。

そして、本書を読み終えた人向けに、本書の公式サイトにさらなるサンプルアプリを用意しています。ぜひ、チャンレンジしてみてください。

まったくの初心者の方は、読み飛ばさずに最初からじっくりと取り組んでみてください。少しでも経験のある方は、作りたいサンプルアプリのレッスンからはじめるのもよいでしょう。

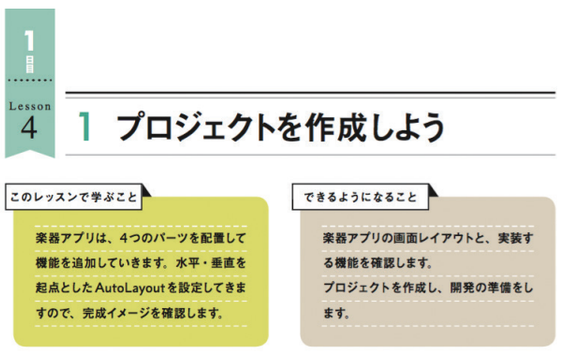
0-4 本書の構成

1日目はレッスン5まであり、iPhoneアプリ制作の概論と開発の準備から入ります。そして、ここで「じゃんけんアプリ」「楽器アプリ」「マップ検索アプリ」の3つのアプリを作ります。「アプリを作って動かすことができた！」という体験を得てください。

2日目はレッスン4まであります。サンプルアプリは「タイマーアプリ」「カメラアプリ（前半）」「カメラアプリ（後半）」「お菓子検索アプリ」を作ります。

0-5 本書の読み方とページ構成

①このレッスンで学ぶこと



レッスンの中で学べることをピックアップして予測できるようにわかりやすくしています。

②プログラムソースコード

1-4 Playボタンの処理を、1-1で作成したsoundPlayerメソッドに差し替え

```
@IBAction func play(sender: AnyObject) {
  soundPlayer(&backmusicPlayer ,path:guitarPath, count: -1)
}
```

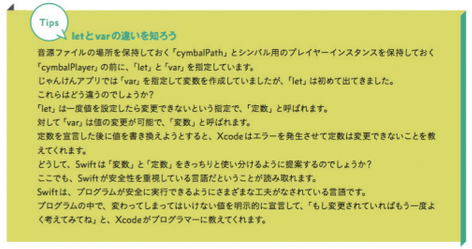
シンバルやギターのコードとの違いは、ループ回数を「count: -1」と指定している点です。
「-1」を指定することで、エンドレスでループ再生をしています。

Swiftのソースコードを掲載しています。ソースコードは理解しやすいように、1行から数行の塊で説明していきます。

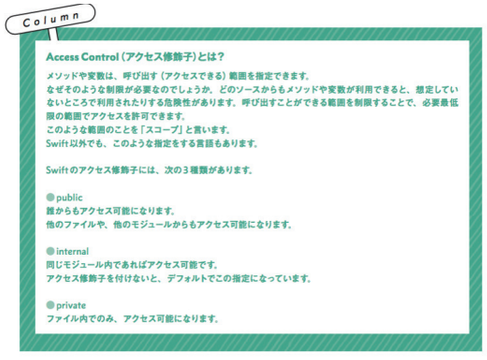
※Swift（スウィフト）は、iOSアプリを作るためのプログラミング言語です。
※Xcode（エックスコード）は、iOSアプリを視覚的に開発するための統合開発環境です。

③Tips（ワンポイント）・Column（技術コラム）

Tips（ワンポイント）は、いまの学習の中での補助的な情報や、知っておいたほうがよいことを記載しています。さらに小さな情報は「MEMO」として掲載しています。



Column（技術コラム）は、本文の流れからは少しそれますが、重要な技術や使い方を記載しています。



0-6 さらなるステップアップ！　～カスタマイズ編について～

本書では、もう一歩進んで学習したい方のためにレッスンごとに「カスタマイズ編」を設けています。

サンプルアプリをさらにカスタマイズして、機能アップさせながら学んでいきます。

カスタマイズ編が難しいと感じられた方は、最初の学習ではカスタマイズ編を飛ばして学習を進めてみてください。2回目以降からはカスタマズ編も含めて学習していくことで、効果的に学ぶことができます。

0-7 本書の公式サポートサイトの紹介とサンプルアプリダウンロード

公式サイトでは、本書の内容に関するサポートや、書籍内で掲載されているサンプルアプリ、プログラムコードなどが提供されています。

完成したサンプルアプリの動きを確認したり、自分で打ち込んだプログラムコードの確認などでご使用ください。

本書の公式サポートサイト

https://swiftbg.github.io/swiftbook/

本書のサンプルアプリダウンロードと使い方

https://swiftbg.github.io/swiftbook/sample/

0-8 本書を読了された方のための応用編講座

本書を最後まで読了された方には、応用編として、さらなる学習教材を用意しています。少し難易度が上がりますが、やりがいがあると思いますので、ぜひ、チャレンジしてみてください！

応用編テキスト・サンプルアプリのダウンロード

https://swiftbg.github.io/swiftbook/advanced/

1：ご利用の前に必ずお読みください

1-1 必要なパソコン機器

iOSアプリ開発には、Macが必要です。

Apple Mac製品

本当にはじめての方は、Windowsパソコンでもアプリ開発ができると思いがちですが、iOSアプリ開発には、**Macが必須**になります。Macであれば、MacBook、iMac、Mac miniのどれであっても大丈夫です。

Mac OSを搭載したパソコンがないと、iOSアプリ開発のための環境を作ることができません。ぜひ、自分に合ったMacの購入を検討してみてください。

1-2 本書で必要な各ソフトウェアのバージョンについて

アプリ開発を行う前に、**必要なソフトウェアのバージョンを確認**して、それを満たす必要があります。バージョンはそのソフトウェアがいつ提供されたものであるのかを示す番号です。

iOSはAppleがMac OSをベースに開発したiPhone、iPad、iPod Touch向けのモバイルOSです。このiOSのバージョンも確認する必要があります。

サンプルプログラムや本書で記載されているプログラムコード、画面掲載は、以下の環境に対応しています。

OS X El CapitanまたはMac OS Sierra
Xcode 8.0以上
iOS10.0以上

上記のバージョンに満たない場合は、バージョンアップを行う必要があります。
バージョンアップに関しては、Appleサポートページをご確認ください。

Appleサポートページ
https://support.apple.com/ja-jp

Tips

アプリ作りの勉強をはじめたときに、最初のハードルとなるのが、Xcode（エックスコード）の操作です。目的の画面を作るためにはXcodeを操作する必要がありますが、最初は、「どの場所にどんな機能のボタンがあるのか」がよくわからないため、なかなか学習が進みません。
そういった問題を克服するために、本書では、どんどん体験して、「体感」で覚えていく構成にしました。ぜひ、サンプルアプリを作りながら「体感」してください！

Lesson

1

はじめてのアプリを開発する前に知っておこう

このレッスンでできるようになること

iOSアプリの開発を始める前に、知っておいた方がよいことを学んでいきましょう。
書籍を読み進めていく上で、最初の心構えを知っておくことで、心理的なハードルが下がり、学習が行いやすくなります。
Appleが開催している開発者向けの大きなイベント「WWDC」についても触れていきます。WWDCでは、Appleの新製品や新機能が発表されたり、市場動向や開発者への支払額などが公開されています。
「Swift」（スウィフト）言語の概要も学んでいきます。Swiftは2014年に発表された新しいプログラミング言語で、Appleが作りました。Swiftは、iPhoneだけでなくMac、Apple Watchでも動くアプリを作ることができます。

1 プログラミングを体験から学んでいこう

このレッスンで学ぶこと

プログラミングに対する気持ちを整理していきます。プログラミングを体験して学んでいくという考え方を理解していきましょう。

できるようになること

プログラミングは難しいものではなく、楽しんで作るものだと理解すると、気負いがなくなります。

1：この本の使い方と前提知識

1-1 体験から学習する

あなたの心の中に、「プログラミングは頭で学ぶもの」という気持ちがあるなら、すぐに切り替えましょう！

プログラミングは「体験しながら体で学ぶ」という方法もあります。

プログラミングの文法や仕組みの理解は、あとからついてきます。

あなたが体験したことが糧となって、徐々に理解できるようになります。

1-2 予習や事前学習について

本書を学習するにあたって、予習のための時間は必要ありません。

予習で最初に知識を詰め込んでも、その知識が必要になるとは限りません。

最初にアプリを作っていき、必要なときに必要な量を学習する方が効率的です。

まずは、実際に作ってみて「動いている！」という成功イメージを持つことが大切です。

1-3 基礎知識、前提知識について

本書では、事前の基礎知識、前提知識はとくに必要ありません。

「動いた！」という体験をしてから、気になること、疑問に思うことを調べて、「なるほど！」と思えたときに基礎知識が身についたといえるでしょう。

「まずは体験してみる」「基礎知識はあとから学習する」ことを念頭において読み進めていくことが大切です。

2 あらかじめ挫折しそうなポイントを押さえておこう

このレッスンで学ぶこと

アプリ開発で挫折しそうなポイントを事前に理解します。本書で集中して学習する、エラーや警告に対する考え方、Xcodeについて学びます。

できるようになること

最初に挫折しそうなポイントに触れているので、心理的なハードルが低くなります。

1：学習ポイントを押さえよう

ポイント①：まずは一冊の本に取り組む

入門書をたくさん購入することで満足してしまいがちですが、まずは1つの書籍を最後まで学習することが大切です。

まずは、じっくりと本書だけに取り組んでみてください。

焦ることはありません。

すべてが理解できなくても、前に進んでいきましょう。

本書を読み終えるころには、次の新しい入門書や文法書も読み進めていくことができるようになります。

ポイント②：アプリ開発をする前の準備

アプリ開発を行う上で、事前の準備が必要になります。

事前の設定ではトラブルも多く、ここで諦めてしまう方も多いと思います。

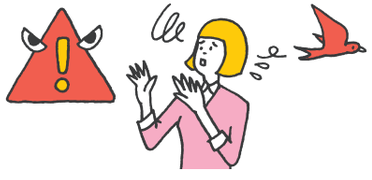
本書では、1日目 Lesson2でアプリ開発の準備にも十分な時間を割いています。事前の準備が終われば、いよいよアプリを作っていきます！

ポイント③：アプリ開発で表示される警告やエラー

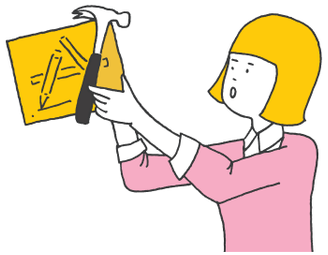
アプリを開発していく過程で、「警告」や「エラー」と言われるものに遭遇することがあります。

最初は、警告やエラーの意味がよくわからず戸惑うことと思いますが、安心してください。警告やエラーはあなたを責めているのではなく、よりよい方法を教えてくれているのです。

警告やエラーに遭遇したときには、「アドバイスしてくれてありがとう！」というぐらいの気持ちをもって、開発に取り組みましょう。



ポイント④：まずは、Xcodeを体験して慣れていこう



アプリ開発はXcodeという統合開発環境を利用して制作を行います。

最初は、Xcodeの操作がよくわからなくてなかなか思うように作業が進みません。

でも、Xcodeでのメニュー配置が理解できて、目的の作業をするための手順がわかるようになると、効率的に制作できるようになります。

Xcodeも理解しようとするよりも体験して、慣れていくことが大切です。繰り返し操作を体験し、ぜひ慣れていきましょう。

3 アプリ開発をするなら知っておこう！ ～WWDC、手数料、課金方法～

このレッスンで学ぶこと

アプリ開発をする上で知っておきたい、WWDCの概要、ダウンロード数などの数値、手数料、課金について概要をつかみます。

できるようになること

WWDCの情報をもとに、アプリ開発の環境を理解していきます。開発者がとても気になる課金や報酬の概要についても知っておきましょう。

1：WWDC2016での公開情報をもとに考察してみよう

1-1 WWDCとは

WWDC（Worldwide Developers Conference）は、iPhoneの開発元であるAppleが毎年開催している、開発者向けのイベントです。

WWDCでAppleの新製品や新機能が発表されたり、市場動向や開発者への支払額が公開されるため、アプリ開発者にとっては、とても関心が高いイベントです。

WWDC - Apple Developer

https://developer.apple.com/wwdc/

WWDC2016は6月13日～17日にサンフランシスコで開催されました。

WWDCはAppleの製品・サービスが発表されるイベントですが、参加するためのチケットは1,599ドル（WWDC2016チケット価格）で販売されていて、1ドル105円換算で約16万7,000円と、その人気の高さがわかります。

WWDCでは、過去に右のような製品・サービスが発表されています。

WWDCでの製品発表

発表年	製品
1998年	iMac
2006年	Intel XserveとMac Pro
2008年	iPhone3G、MacBook Air
2010年	iPhone4、iPad
2012年	MacBook Pro（Retinaディスプレイ搭載）
2013年	Mac Pro（円筒形デザイン）
2014年	Swift
2015年	Swift2.0、Swiftオープンソース化
2016年	Swift3.0、Sirikit、iMessage Apps

1-2 App Storeのダウンロード数と開発者への支払額

アプリ開発者は平均でどのぐらいの収入を得ているのでしょうか？

WWDC2016での公式発表の数値から試算してみたいと思います。

WWDC2016での公式発表によると、App Storeからのアプリダウンロード数は1,300億本を超えており、2008～2016年までの8年間でAppleが開発者に支払った金額は累計500億ドルです。累計額を1ドル105円換算すると、約5.3兆円にもなります。また、登録アプリ数は約200万本であるとされるため、この5.3兆円を200万本のアプリ数で割ると、1つのアプリの平均収入は約265万円になります。

2008年からアプリを1つリリースしていたとすると、年間平均で約33万円、月平均で約2万7千円の収入になります。

そして、登録アプリ数の200万本には無料アプリも含まれていますので、平均は約2万7千円以上になると思われます。

1-3 Appleの手数料

Apple Storeで販売した売上のうちの30%をAppleが手数料として差し引きます。これがAppleの収益になります。残りの70%が開発者への支払い額です。支払いは、指定した銀行口座に振り込まれます。

たとえば、120円のアプリであれば、84円が振り込まれることになります。

1-4 ユーザーへの課金方法

iPhoneアプリをApp Storeに公開してユーザーに課金する方法は3種類あります。

ユーザーへの課金方法

課金方法	説明
有料ダウンロード	アプリをダウンロードするときに課金する方法。ユーザーは1度ダウンロードすれば、その後のアップデートなどは無料で実行可能。アプリを削除して再びダウンロードするときも無料
広告	アプリの中に広告を表示させて収益を得る方法。いろいろな会社がアプリ向けの広告配信サービスを提供している
In App Purchase（アプリ内課金）	アプリ内で課金する方法。有料もしくは無料でアプリを提供し、そのアプリ内でアイテムや機能追加を販売し、収益を得る。アプリ内課金には、次の4つの方法がある ①消耗型：アプリの実行に伴い消費されていく。消費アイテムなど ②非消耗型：1度購入するとユーザーのすべてのデバイスで使用可。書籍など ③自動更新購読：期間を決めて販売、自動で更新。新聞、雑誌など ④非更新購読：期間を決めて販売、自動で更新されない。1ヶ月購読など

4 Swift（スウィフト）を知ろう

このレッスンで学ぶこと

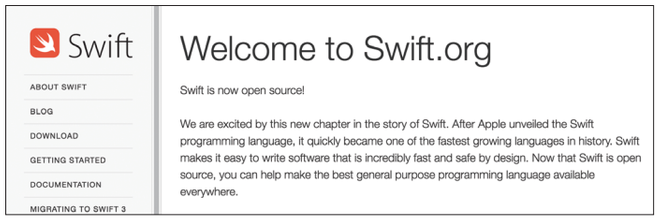
iOSアプリ開発では、Swift（スウィフト）というプログラミング言語を使います。2014年に登場した新しい言語です。Swiftの概要を理解していきます。

できるようになること

Swiftには高速、モダン、安全、インタラクティブといった特徴があります。Swiftを学ぶにつれてこれらのことがよくわかるようになります。

1：Swiftの特徴を押さえよう

みなさんが、iOSアプリを作るために使うプログラミング言語がSwift（スウィフト）と呼ばれるものです。Swiftは2014年のWWDCではじめて発表されました。



Swift.org - Welcome to Swift.org

https://swift.org/

Appleの公式サイトでは、Swiftとは「誰もが圧倒的に優れたアプリケーションを作れる、パワフルなオープンソースの言語」として紹介されています。

Swiftを使うことでアプリ開発者はより安全で、より信頼性の高いコードを書くことができ、時間を節約しながら、より豊かなアプリを作ることができます。

●特徴①：Swiftは高速である

Swiftは日本語訳で「迅速」という意味で、鳥の「あまつばめ」を示す意味でも使われます。

Swiftの迅速さを強調するために、ロゴマークにも「あまつばめ」が採用されています。

Swiftは他のプログラミング言語と比較して、検索アルゴリズム（データを探し出す方法）が高速だと言われています。

●特徴②：Swiftはモダンである

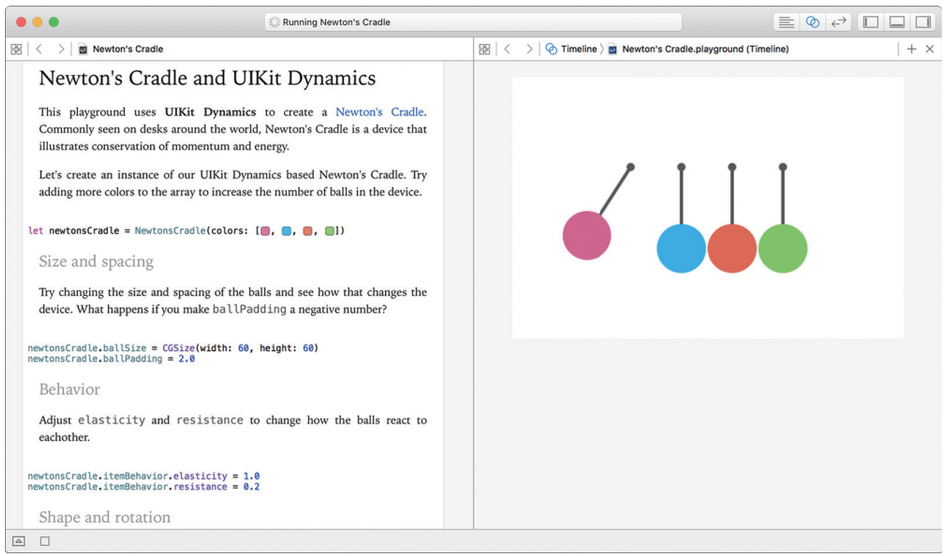
モダン（modern）とは「現代風」という意味です。Swiftでは、他のプログラミングでも採用されている新しい機能を、積極的に取り入れています。

●特徴③：Swiftは安全である

Swiftはプログラミングの記述ミスやバグ（不具合）が起こりにくい仕組みを採用しています。

●特徴④：Swiftはインタラクティブである

Swiftはインタラクティブ（対話形式）に動かすことができます。



Playgrounds

Playgrounds（プレイグラウンド）と呼ばれる、Swiftを記述するとリアルタイムで実行結果がわかる機能があり、Swiftの動作を確認する際にとても便利です。

●特徴⑤：たくさんのApple製品で動くアプリが作れる

Swiftで作れるアプリはiPhoneやiPad上で動くアプリだけではありません。Mac、Apple Watchなどたくさんの Apple製品で動くアプリを作ることができます。

●特徴⑥：Swiftはオープンソースである

オープンソース（Open Source）とは、プログラムソースを一般に公開して、誰もが使ってよいとする考え方です。

オープンソースであれば、一般人もSwiftの改良に参加することができますので、ネット上では日々意見交換され、さらなる改良・進化していくことが期待されています。

Tips

Swiftが発表される前は、「Objective-C」（オブジェクティブシー）というプログラミング言語を使って、アプリ開発を行っていました。
Objective-Cは1980年代から開発がはじまり、機能拡張を経て現在も使用されています。Objective-Cは、記述が長くなり複雑であったり、プログラムを書いていく効率があまりよくなかったりと、初心者には難しい言語です。
Swiftでは、学びやすいような工夫がされていて、新しい考え方や機能を積極的に取り入れています。
Swiftもバージョンアップを重ねて、十分に開発が行える言語として成長しています。
また、過去の資産を有効活用できるように、Objective-Cで作成されたプログラムをSwiftから呼び出すこともできます。
本書は、これからの開発言語である「Swift」でプログラムを記述していきます。

Lesson

2

アプリ開発の環境を整えて、Xcodeの使い方を学ぼう

このレッスンでできるようになること

iOSアプリを開発するために、必要なものを学んでいきます。
Apple IDを取得されていない方のために、Apple IDの説明と取得方法についても解説します。
そして、アプリ開発をするために必要なXcode（エックスコード）のインストールを学び、iPhoneの画面に「Hello Swift!」と表示できるようなアプリを作ってみます。
最初の簡単なアプリを作りながら、Xcodeにどのような機能があるのか体験していきましょう。
基本の操作を行いながら、iPhone（実機）がなくてもテストが行えるシミュレータの操作や、iPhone本体へのアプリ転送（実機転送）も学びます。

日目

1 iOSアプリを開発するために必要な準備をしよう

このレッスンで学ぶこと

iOSアプリを開発するために、用意しなければならないものや、それぞれの役割について学んでいきます。

できるようになること

iOSアプリ開発をはじめる前に必要な知識である、MacやApple ID、Xcodeの概要について理解していきます。

1：アプリ開発に必要な3つのものを準備しよう

iOSアプリを開発するためには、次の3つが必要になります。

Mac
Apple IDアカウント
Xcode

また、開発したアプリを全世界の人たちに利用してもらえるようにするには、別途、**「Apple Developer Program」**への登録も必要です。

この章では開発したアプリを、iPhoneに転送（実機転送）を行って利用するところまでをゴールにしています。

アプリを開発して世界に公開したいという方は、本書の公式サイトで公開手順を掲載していますので、確認してみてください。

アプリの公開手順
https://swiftbg.github.io/swiftbook/release

それでは、必要な3つのものを確認していきましょう。

1-1 Mac

2015年12月に、Swiftは誰でも利用・改変できるオープンソースとしてソースコードが公開されました。そのため、Mac以外のパソコンでもiOSアプリ開発ができる可能性もありますが、本書を執筆している2016年9月時点ではMacが必要になります。

現時点ではMacを準備してください。

Mac

1-2 Apple ID

iOSアプリを開発するためには、**Apple ID**の作成が必要です。

Apple IDは、Appleが提供しているオンラインサービスを利用するために必要なアカウントです。

オンラインストアの「iTunes Store」では、音楽・映画やオーディオブックを購入できます。

また、「App Store」というサービスでは、iPhoneやiPad、Macで利用できるアプリがダウンロードできます。さらに、有料で販売されているアプリも購入することができます。

App StoreとiTunes

iPhoneやiPadをお持ちの方は、すでにApple IDを利用していると思います。

それぞれのサービスを利用するのに必要な**Apple ID**ですが、Xcodeのダウンロードでも必要になるため、事前に作成しておく必要があります。

1-3 Xcode（エックスコード）

Apple IDが作成できたら、「App Store」で、iOSアプリ開発に必要なXcodeをインストールします。

Xcodeは、Mac、iPhone、iPad、Apple Watch向けのアプリを開発できる環境を提供してくれます。

Xcodeを利用して、画面やコードの作成、デバッグ、App Storeへのアプリの提出ができます。

一般的にこのようなツールは、**統合開発環境**もしくは、**IDE（Integrated Development Environment）**と呼ばれています。

Xcode

では、**IDE**とはなんなのでしょう？

IDEとはソフトウェアを効率よく開発できるように、さまざまな機能を提供してくれている開発ツールです。

SwiftやObjective-Cを使って開発ができるXcode以外にも、さまざまな言語（Java、Ruby、PHPなど）で、有料無料問わずにたくさんのIDEがあります。

また、Xcode以外にもiOSアプリ開発ができるIDEはありますが、一般的に多く利用されているのはXcodeです。

本書でも、Xcodeをインストールして開発を進めていきます。