

はじめに

本書は、スマートフォン向けアプリケーション開発ツール「PhoneGap/Cordova」を使い、iPhone向けのアプリケーションを開発するために必要な知識を基礎から解説した入門書です。

● 本書の対象読者

- ・はじめてWebアプリケーションを製作する初心者
- ・iPhoneアプリを開発してみたいWebプログラマ、Webデザイナー
- ・PhoneGapを使って開発効率向上を目指す、iPhoneアプリ開発者

本来スマートフォン向けの開発では、各OSごとに特定の言語を使用して開発を行う必要がありました。たとえばiOSデバイスの場合は、Objective-Cを習得する必要があります。

プログラミングの初心者がiPhone向けアプリを開発できるようになるには、難易度の高いものでした。

PhoneGapが登場したことで、Webアプリケーションを開発する要領でiPhone向けアプリの開発が可能になりました。

本書では、PhoneGapの導入から、HTML5+JavaScriptとPhoneGapで作成するサンプルをベースに「アプリケーション開発にあたり、外せないポイント」「はまりやすいポイント」を一つ一つ段階を追って解説していきます。

また、進化の早い技術に追従できるように「基礎的な知識」と「追いかけるべき情報源」を網羅しました。

現場で活躍するWebプログラマ、Webデザイナーはもちろん、これまでWebアプリケーションを作ったことがない方にも呪文のような長い英文を丸覚えするのではなく、「どうしてそうなるのか」を整理しながら理解し、楽しんで開発してもらえるように工夫を凝らしました。

本書が読者の皆様にとって、iPhoneアプリケーション開発の楽しさを知るきっかけになれば幸いです。

2012年8月

富田宏昭

CONTENTS

CHAPTER_01

PhoneGapについて 011

#01-01 PhoneGapとは 012

▶▶▶ PhoneGapを開発に使う意味 012

▶▶▶ PhoneGapの歴史的背景 013

#01-02 必要な開発環境 015

▶▶▶ iOS SDKとは 015

▶▶▶ Xcodeについて 016

▶▶▶ iOS Developer Programとは 016

▶▶▶ PhoneGapのインストール 017

CHAPTER_02

開発環境のインストール 019

#02-01 iOS SDKのインストール 020

▶▶▶ iOS Dev Centerへの登録 020

▶▶▶ iOS SDKインストーラのダウンロード 023

▶▶▶ インストーラの起動・実行 024

▶▶▶ Xcode 4 の起動方法 025

TIPS Mac OS Xのバージョンについて 024

TIPS Xcodeの古いバージョンをインストールしている場合 026

#02-02 PhoneGapのインストール 027

▶▶▶ PhoneGapのダウンロード 027

▶▶▶ インストーラの起動・実行 028

TIPS PhoneGapとCordova	028
TIPS OS X 10.8でのPhoneGapのインストールについて	028
#02-03 プロジェクトの作成方法	030
▶) Xcode 4 起動 ~ Cordovaプロジェクトの作成	030
▶) 作成するアプリケーションの決定	034
TIPS その他のプロジェクト作成方法	030
TIPS Bundle Identifier	032
TIPS Organization Nameについて	032
TIPS PhoneGap1.9でiOS5.0以下のアプリを作る時の注意点	032
TIPS ショートカットキー	040
Column 01 Xcode 4 の画面構成と操作の基本	041
Column 02 iOS シミュレータの使い方	044
#02-04 PhoneGap 2.0.0での変更点	049
▶) PhoneGap 2.0.0でプロジェクトを作成するには	049
▶) PhoneGap 2.0.0でのファイル構成の変更点	054
▶) PhoneGap 2.0.0での機能変更点	054
TIPS Xcodeテンプレートを使わなくなった理由	049
TIPS createコマンドについて	052
TIPS ターミナル操作について	053

CHAPTER_03

HTML+JavaScriptで作る アプリケーション

#03-01 HTML/CSS/JavaScriptについて	056
▶) HTMLとは.....	056

▶ CSSとは	062
▶ JavaScriptとは	067
TIPS HTML5を使うと、何がすごいのか?	062
TIPS RGBとは	065
#03-02 JavaScriptの基本文法とDOMについて	070
▶ JavaScriptの基本文法	070
▶ DOMとは	075
TIPS 変数名について	070
TIPS セミコロン(;)について	072
03-03 JavaScriptの基本を学びながら「カウンタ」のアプリを作ろう	084
▶ プロジェクト作成とコードの記述	085
▶ 実際に動作させてみる	088
▶ エラーが起きたら	089
▶ アプリケーションの解説	090
TIPS iOS Human Interface Guidelines	097
#03-04 変数と配列を学びながら「くじびき」のアプリを作ろう	098
▶ プロジェクト作成とコードの記述	099
▶ 実際に動作させてみる	100
▶ アプリケーションの解説	101
TIPS エスケープシーケンス	106
TIPS シングルクォーテーションとダブルクォーテーション	109
#03-05 制御構文を使って「かけ算」のアプリを作ろう	110
▶ プロジェクト作成とコードの記述	111
▶ 実際に動作させてみる	113

▶) アプリケーションの解説	114
▶) ローカル変数とグローバル変数について	128
Column 03 用語について	133
Column 04 デバッグについて	136

CHAPTER_04

PhoneGapで作るアプリケーション 基本編

.....	137
#04-01 PhoneGap/Cordovaでアプリを作るには	138
▶) Cordovaでアプリを作るには	138
#04-02 Cordova APIの紹介	141
▶) Cordova APIとは	141
▶) Cordova API一覧	141
TIPS Base64エンコード	143
#04-03 Storage APIを使って「簡易メモ帳」のアプリを作ろう	147
▶) プロジェクト作成とコードの記述	148
▶) 実際に動作させてみる	149
▶) アプリケーションの解説	151
▶) 応用編：複数のメモを作成できるようにしよう	156
▶) コードを記述する	157
▶) 実際に動作させてみる	161
▶) アプリケーションの解説	163
#04-04 タイマー処理を使って「キッチンタイマー」のアプリを作ろう	174
▶) プロジェクト作成とコードの記述	175
▶) 実際に動作させてみる	177
▶) アプリケーションの解説	179

▶▶▶ 応用編：タイマー完了時に音声ファイルを再生する	189
▶▶▶ コードを記述する	190
#04-05 Compass APIとCanvasを使って	
「コンパス」のアプリを作ろう	192
▶▶▶ プロジェクト作成とコードの記述	193
▶▶▶ 実際に動作させてみる	196
▶▶▶ アプリケーションの解説	197
#04-06 Geolocation APIとGoogle Maps APIを使って	
「地図」のアプリを作ろう	211
▶▶▶ プロジェクト作成とコードの記述	212
▶▶▶ ExternalHostの追記	214
▶▶▶ 実際に動作させてみる	218
▶▶▶ アプリケーションの解説	219
▶▶▶ 応用編：移動した経路を地図に表示しよう	226
▶▶▶ 実際に動作させてみる	228
▶▶▶ アプリケーションの解説	229
TIPS ホワイトリストとブラックリスト	217
TIPS 同期処理と非同期処理の違い	218
#04-07 Camera APIを使って「カメラ」のアプリを作ろう	233
▶▶▶ プロジェクト作成とコードの記述	234
▶▶▶ 実際に動作させてみる	236
▶▶▶ アプリケーションの解説	237
TIPS データURLスキーム	242
Column 05 OSバージョン判別式について	244
Column 06 回線情報の特定について	246
Column 07 Events API について	247
Column 08 Cordova Pluginについて	248

CHAPTER_05

PhoneGapで作るアプリケーション 応用編

.....	249
#05-01 「jQuery」とCordova Contacts APIを使って	
「電話帳移行アプリ」を作ろう	250
▶ プロジェクト作成とコードの記述	252
▶ iTunes 経由でファイルを転送できるようにする	260
▶ 実際に動作させてみる	263
▶ アプリケーションの解説	266
TIPS \$について	266
TIPS 無名関数について	268
TIPS Ajaxとは	272
TIPS ドキュメント構造について	273
TIPS アプリケーション本体と「Documents」「tmp」ディレクトリについて	273
#05-02 外部ライブラリとFile APIを使って	
画像を加工するアプリを作ろう	294
▶ プロジェクト作成とコードの記述	295
▶ iTunes 経由でファイルを転送できるようにする	307
▶ 実際に動作させてみる	307
▶ アプリケーションの解説	309
TIPS プラグインのアップデートについて	304
TIPS FileEntryオブジェクト	319
TIPS 使いたい機能の詳細がマニュアルに書いていない場合	323

CHAPTER_06

PhoneGap開発TIPS 325

#06-01 実機で動作確認するための方法 326

▶ iOS Developer Programへの登録 326

▶ プロビジョニングファイルの登録手順 329

▶ デバイスにプロビジョニングファイルを関連づける 336

#06-02 PhoneGap:Build の利用方法 339

▶ PhoneGap : Buildとは 339

▶ PhoneGap : Buildを利用する 342

TIPS バージョン管理システム 344

#06-03 JavaScriptライブラリについて 350

▶ JavaScriptライブラリの利点 350

#06-04 Cordovaアップデート方法について 353

▶ Cordova 1.x.xから2.0.0へのアップデート 353

▶ 既存のCordovaプロジェクトのアップデート方法 353

▶ プラグインのアップデート時における注意点 357

▶ 最新版にアップデートする際の注意点 358

▶ チェックしておきたいWebサイト 358

#06-05 iOS Developer License更新手順 361

▶ iOS Developer Licenseは更新が必要 361

INDEX 365



PhoneGapとは、Adobeが公開しているクロスプラットフォーム、モバイルアプリケーションの開発フレームワークです。開発者は無料で利用することができます（2012年8月現在）。まずはこのPhoneGapで何ができるのかを、そして生き立ちを紹介しましょう。

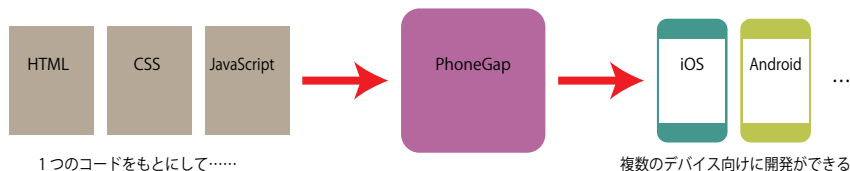
📞) PhoneGapを開発に使う意味

本来スマートフォン向けの開発では、各OSごとに特定の言語を使用して開発を行う必要がありました。たとえばiOSデバイスの場合はObjective-C、AndroidやBlackBerryの場合はJavaといった言語で開発を行います。ですから、複数のスマートフォンに対応するアプリケーションを開発する場合、デベロッパは複数の言語と開発環境を用意して、コーディングを行って開発を進めていくことになります。

Objective-CやJavaは、プログラマならともかく、WebデザイナーやWebプログラミングの初心者学習するには難易度の高い言語です。開発環境の準備や、開発環境そのものの学習、言語の学習期間などを考慮した場合、すぐに始めるには相応の労力と時間を確保する必要が出てきてしまいます。

しかし、PhoneGapが登場したことで、プログラマのみならず、「何回かWebサイトを作成したことがある」といった初心者まで、簡単にスマートフォン向けのネイティブアプリケーションを開発できるようになりました。

PhoneGapをもちいることで、HTML+CSS+JavaScriptのみでスマートフォン向けのネイティブアプリケーションを開発することが可能になります。つまり、Webアプリケーションを作成する要領でスマートフォン向けアプリケーションを開発できるのです。さらに、JavaScriptに用意されているPhoneGapライブラリを使って、カメラやGPS、電話帳といったデバイスの機能をも操作することができます。



Webアプリケーションを作る要領で、iOS/Androidの機能を活かしたネイティブアプリケーションが作成可能

PhoneGapを使うと、基本的な部分は1ソースで、複数のスマートフォン向けアプリケーションを作ることができます。たとえばiOS向けに作ったアプリケーションのコードを、少しの変更でAndroid OS向けにすることもできるのです。

ただし、現状では、iOS向けのアプリケーションを作ると、Android向けのアプリケーションを作るのでは、それぞれ異なる環境を作らなければなりません。iOS/Androidのバージョンごとに画面デザインを用意しなければならないといった細かい部分での微調整などにも手間がかかります。

対象OSごとに必要な開発環境			
対象OS	必要なSDK	開発環境	プラットフォーム
iOS	iOS SDK	Xcode	Macintosh
Android	Android SDK	Eclipseなど	Macintosh/Windows

本書では、初心者の方でもスマートフォン向けのアプリケーションを作れるように、iOS向けのアプリケーションに絞って解説を行っていきます。

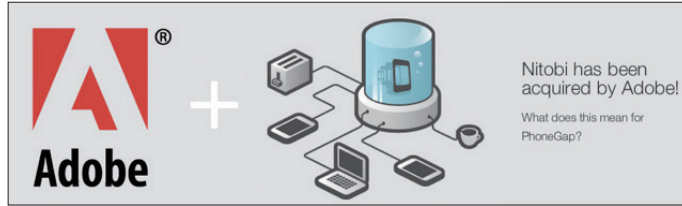
PhoneGapを用いて作成したiOS向けアプリケーションは、iPhone/iPadのApp Storeにて公開することができます。ただし、App Storeで公開するには、Appleの審査を通過する必要がありますので、この点は注意が必要です。App Storeでの公開は、アプリの値段としては無償/有償どちらでも可能ですし、社内向けのツールとして利用するなど、さまざまな活用方法が望めます。

🔊 PhoneGapの歴史的背景

もともとPhoneGapは、Nitobi Software社（以下Nitobi）が開発、公開していました。2008年8月にサンフランシスコで開かれたiPhoneDevCampをきっかけに開発がはじまりました。その後、機能の追加や改修を重ね、去る2011年7月28日（米国時間）にPhoneGap 1.0.0がリリースされる運びとなりました。

そして同年10月2日（米国時間）、Adobe Systems社（以下Adobe）が開催するAdobe MAX 2011にて、AdobeがPhoneGapの開発元であるNitobiを買収したと発表しました。もともとAdobeの製品である「Dreamweaver」においてPhoneGapのサポートがすすめられていましたが、この買収を持ってAdobe社が実権を握ることとなりました。PhoneGapはその開発体制や配布方法が変わり、Apacheに寄贈されました。

| ☒ | Nitobiの買収



<http://phonegap.com/>より引用

AdobeがPhoneGapの商標を握ったことで、PhoneGapという名称はAdobe以外使用できなくなりました。Apacheに寄贈されたソースコードは、Apache Callbackプロジェクトとなります。程なくして名称が変更され、Apache Cordovaプロジェクトとなりました。現在、PhoneGapはCordovaのディストリビューション（配布形態）の1つとして公開されています。

PhoneGapプロジェクトを率いているBrian LeRoux氏は、2012年3月19日（米国時間）に公開されたブログ記事^(※)にてPhoneGapとCordovaの関係を次のように語っています（一部を抜粋して翻訳）。

“PhoneGapとCordovaの関係は、Google ChromeやSafariといったWebブラウザのレンダリングエンジンが、WebKitである関係と似たようなものです。PhoneGapのエンジンはCordovaになります。”

(※) <http://phonegap.com/2012/03/19/phonegap-cordova-and-what%E2%80%99s-in-a-name/>

このため、PhoneGapのインストーラはCordovaという名前がついています。また、PhoneGap公式Webサイトの開発用ドキュメントにもPhoneGapの文字はなく「Apache Cordova Documentation」となっています。本書ではWebサイトや開発用ドキュメントの表記の関係上、PhoneGapとCordovaを使い分けて表記している箇所がありますが、あまり深く考えずに読み進めていただければと思います。

PhoneGapの原稿執筆時の最新バージョンは 2.0.0 です。PhoneGapはApache License, Version 2.0のもとで、PhoneGap公式サイト（<http://phonegap.com/>）およびGitHub（<https://github.com/phonegap/phonegap>）にて公開されています。

ただし、PhoneGap2.0ではXcode用のテンプレートがなくなり、コマンドラインからのプロジェクト作成が必須となりました（49ページ参照）。初心者を対象として考えた場合、コマンドラインの使用は敷居が高いので、本書ではPhoneGap1.9をメインに使用していくことにします。

iOSアプリケーションの開発には、iOS SDKと呼ばれるアプリケーション開発キットが必要になります。iOS SDKは統合開発環境のXcodeに同梱されています。ここではiOS SDK、Xcode、および開発者ライセンス「iOS Developer License」などについて紹介します。

🔊 iOS SDKとは

iOS SDK(Software Development Kit)は、iOSアプリケーションを開発するためのツールキットです。iPad、iPhone、iPod touch向けアプリケーションを作成するための開発ツールがすべて内包されています。iOSアプリケーションを開発するには、まずこのiOS SDKをインストールする必要があります。



Xcode and the iOS SDK.

iPhone/iPod touch/iPad向けに素晴らしいアプリケーションの開発を始めるために必要なすべてを入手できます。Xcodeおよび、Xcode IDE、iOS Simulator、Instruments、Interface Builderなどを含むiOS 5 SDK。

今すぐダウンロード

- 📄 iOS 5の最新機能
- 📄 iOS App Programming Guide
- 📄 iOS Human Interface Guidelines
- 📄 Creating a Universal App



<http://developer.apple.com/jp/ipad/sdk/>より引用

iOS SDKに含まれているおもなアプリケーションは次のとおりです。

- ・Xcode IDE: 統合開発環境
- ・iOS Simulator: PC上で動作するiPhone/iPadエミュレータ
- ・Instruments: パフォーマンス測定ツール。CPU負荷やメモリークを解析
- ・Interface Builder: アプリケーションのUIを簡単に作成するためのツール

iOS SDKにはバージョンがあり、開発環境として利用するMac OS Xのバージョンや、動作環境のiOSバージョンによってダウンロードするバージョン、インストール方法が異なります。

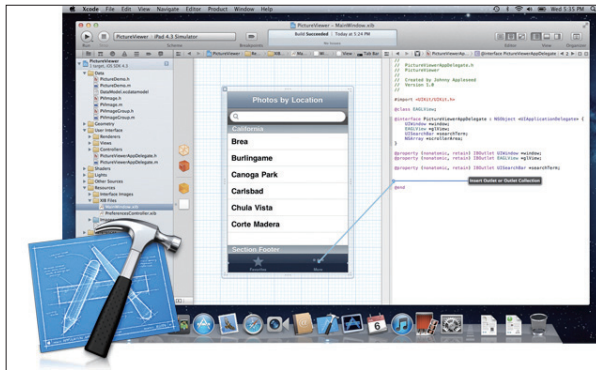
執筆時点の場合、開発環境としてMac OS 10.6を利用している場合は、Xcode 4.2をiOS Dev Centerからダウンロードしてインストールします。Mac OS 10.7を開発環境として利用している場合は、Xcode 4.3または4.4をApp Storeからダウンロードしてインストールすることになります。Mac OS 10.8を利用している場合は、Xcode4.4をインストールします。iOS SDKのインストールについては、「02-01 iOS SDKのインストール」で説明します。

🔊 Xcodeについて

XcodeはMac OS X上で動作する統合開発環境(IDE)です。プロジェクト管理、コードの編集、デバッグ、ファイルの管理をこのXcodeひとつですべてを行うことができます。執筆で使用したXcodeのバージョンは、Xcode 4.3.3です。

Xcodeの起動や操作方法については、「02-01 iOS SDKのインストール」と、「Column 01 Xcode 4の画面構成と操作の基本」で説明します。

🖼️ Xcode



<http://developer.apple.com/technologies/ios/>より引用

🔊 iOS Developer Programとは

作成したiOSアプリケーションをiPhoneシミュレータ上だけで動作させる場合には、iOS Developer Programへの参加は必要ありません。iPhoneシミュレータ上ではカメラや傾きセンサといった一部のデバイスの機能が利用できません。

作成したiOSアプリケーションをiOSデバイスへインストールしたり、App Storeにて公開・配布するためにはiOS Developer Programに参加する必要があります。

iOS Developer Program
The fastest path from code to customer.

今年で登録 年間参加費 ¥8,400

1. 開発
iOS SDKとiOS Dev Centerの豊富なテクニカルリソースを使ってアプリケーションを開発します。さらに詳しく >

2. テスト
アプリケーションの仕上げには、iPad / iPhone / iPod touch上でコードの検証やアプリを実行します。さらに詳しく >

3. 配布
App Store経由でアプリケーションを配信して、何百万人にもiPhone / iPad / iPod touchユーザーに届けられます。さらに詳しく >

<http://developer.apple.com/jp/ipad/sdk/>より引用

個人向けのiOS Developer Program年間参加費は 8,400円です。企業向けのiOS Developer Enterprise Program参加費は 24,800円です(2012年8月現在)。なお金額は為替変動によって変更される場合があります。企業内で使う、社内業務向けのアプリケーションを開発するためには、iOS Developer Enterprise Programへの参加が必要です。

iOS Developer Programへの登録方法は、「06-01 実機で動作を確認するための方法」で説明します。

🔊) PhoneGapのインストール

PhoneGapを使ったiOSアプリ開発を行うには、もちろんPhoneGapのインストールも必要です。PhoneGapを使ったiOSアプリ開発では、PhoneGapをインストールした状態のXcodeを使って、開発を進めていくことになります。

PhoneGapのインストールについては、「02-02 PhoneGapのインストール」で説明します。



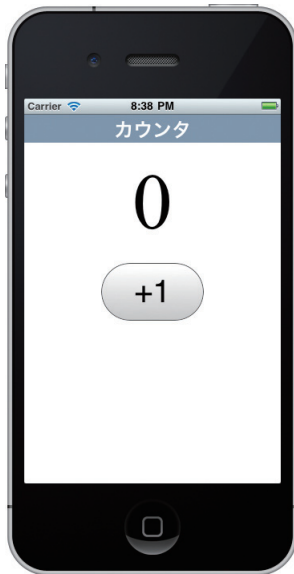
JavaScriptの基本を学びながら 「カウンタ」のアプリを作ろう

まずは簡単に作れるアプリケーションから、HTMLとJavaScriptの基本を押さえていきましょう。

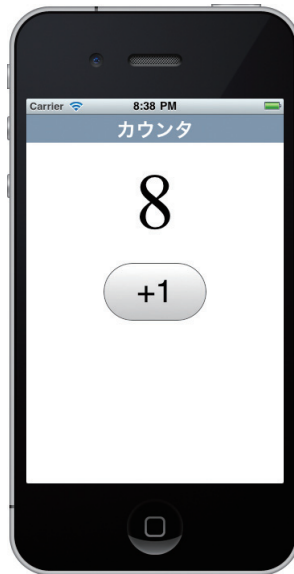
ここで作るアプリについて	
アプリ名	カウンタ
説明	ボタンをクリックするとカウンタが +1 ずつ上がっていく
プロジェクト名	Counter

このアプリの用途：

カウンタを作成します。画面上に表示されているボタンをタップすると、カウンタが1、2、3…と+1ずつカウントアップしていきます。



「+1」のボタンをクリックすると…



画面上の数字が1ずつ上がっていきます

🔊 プロジェクト作成とコードの記述

プロジェクトを作成し、開発を開始しましょう。作成の手順は「02-03 プロジェクトの作成方法」で解説したとおりですが、再度おさらいの意味を込めて確認していきましょう。

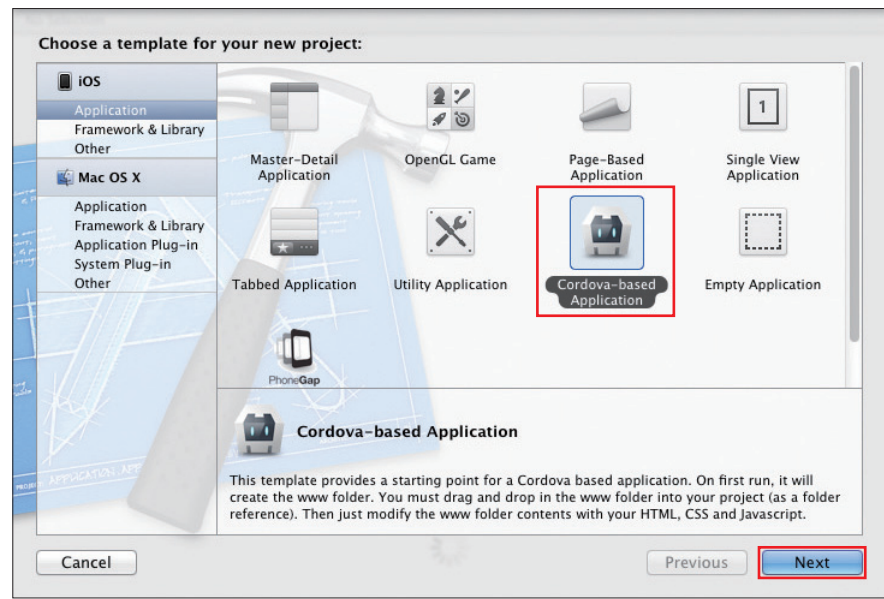
STEP_01

Xcodeを起動し、[Create a new Xcode project] をクリックして、プロジェクトを作成します。



STEP_02

プロジェクトのテンプレート一覧より [Cordova-based Application] を選択し、[Next] ボタンをクリックします。



STEP_03

作成するプロジェクトの「Product Name」と、「Company Identifier」を入力します。ここでは、次の内容を入力します。

Product Name : 「Counter」

Company Identifier : 「jp.mynavi.www」

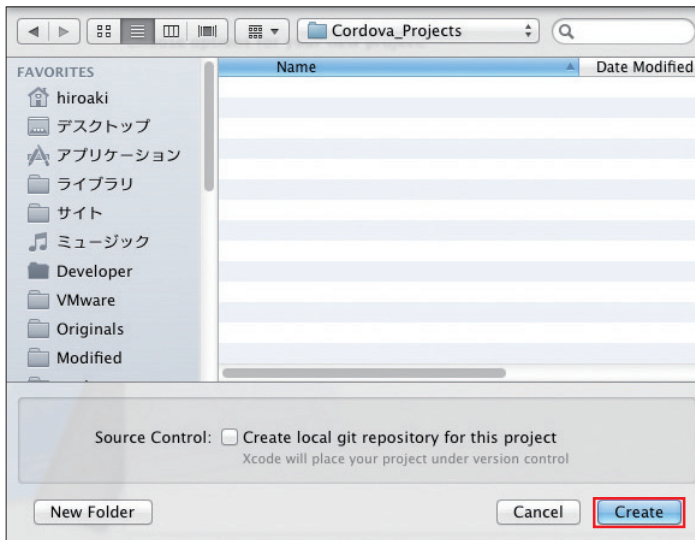
Product Name: Counter
 Company Identifier: jp.mynavi.www
 Bundle Identifier: jp.mynavi.www.Counter
 Use Automatic Reference Counting

「Product Name」と「Company Identifier」を入力後、「Use Automatic Reference Counting」のチェックが外れていることを確認し、「Next」ボタンをクリックします。

STEP_04

プロジェクトを保存するディレクトリを指定します。「Create」ボタンをクリックして、

プロジェクトの作成完了です。プロジェクト「Counter」を作成しました。

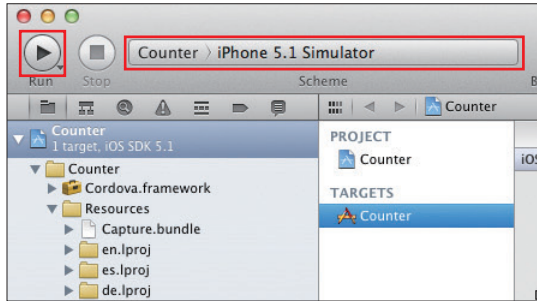


STEP_05

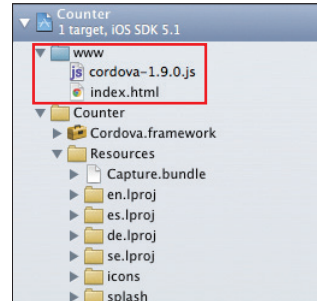
Xcode IDE左上の [Scheme] にて [iPhone 5.1 Simulator] を選択し、[Run] ボタンを押して一度プロジェクトを実行します。

その後、iOSシミュレータ上で**ERROR: Start Page at 'www/index.html' was not found.**エラーが発生していることを確認します。

プロジェクト名「Counter」を右クリックし、[Show in Finder] を選択します。Finderに表示されている [www] フォルダを、Xcodeのプロジェクト名(この場合はCounterの上)にドラッグ&ドロップし、パスを通します。パスの通し方について詳しくは、38ページを参考にしてください。



Schemeを [iPhone 5.1 Simulator] に変更し、[Run] ボタンをクリック



パスを通した後のXcodeのファイルツリー

STEP_06

それではよいよプログラミング開始です。コードを書き換えます。コードは次のとおりです。左側でwww/index.htmlをクリックして開き、

| スクリプト | サンプルファイル: 03/Counter/www/index.html

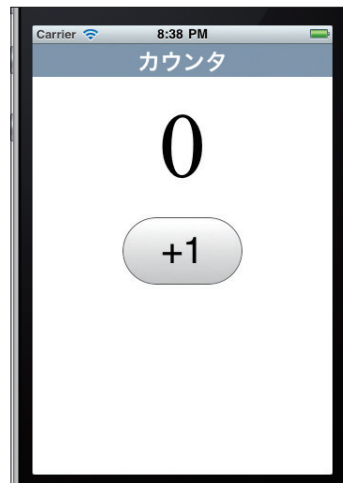
```
01 <!DOCTYPE html>
02 <html>
03   <head>
04     <title>Counter Example</title>
05     <meta charset="utf-8">
06     <meta name="viewport" content="width=device-width, initial-scale=1.0,
07       maximum-scale=1.0, user-scalable=no;" />
08     <script type="text/javascript">
09       window.addEventListener('load', init, false);
10       function init() {
11         document.getElementById('countUp').addEventListener('click', countUp,
12           false);
13       }
14       function countUp() {
15         document.getElementById('counter').textContent =
16           parseInt(document.getElementById('counter').textContent) + 1;
17       }
18     </script>
19     <style type="text/css">
```

```
20 body {
21     margin: 0;
22     text-align: center;
23 }
24 h1 {
25     margin: 0;
26     background-color: #889bb3;
27     color: #ffffff;
28     font-size: 18pt;
29 }
30 #counter {
31     font-size: 74pt;
32 }
33 #countUp {
34     font-size: 30pt;
35 }
36 </style>
37 </head>
38 <body>
39     <h1>カウンタ</h1>
40     <p><span id="counter">0</span></p>
41     <p><button id="countUp">+1</button></p>
42 </body>
43 </html>
```

🔊) 実際に動作させてみる

STEP_01

それでは実際にカウンタアプリケーションを動作させてみましょう。[Run] ボタンをクリックします。エラーがない場合は、iPhoneシミュレータが起動し、作成したカウンタアプリケーションが表示されます。



STEP_02

シミュレータ上のボタンをタップして、カウンタが+1ずつ繰り上がっていくか確認してみましょう。
 無事にカウントアップしていきませんか？
 iPhoneシミュレータが起動しなかったり、ボタンをタップしてもカウンタが動かない場合は次の箇所をチェックしましょう。



🔊) エラーが起きたら

記述したプログラムにミスがある場合や、必要なファイルが足りていない場合、ビルドに失敗することや、アプリケーションが起動してもうまく動作しないケースがあります。

||||| ビルドに失敗、iPhoneシミュレータが起動しない |||

次の3点を確認してみましょう。

- ・プロジェクト作成時に [User Automatic Reference Counting] のチェックを外したか
- ・Schemeが正しく選択されているか
- ・wwwディレクトリの下に、Counter、Frameworks、Productsライブラリー式があるか

まず [Cordova-based Application] テンプレートを使ってプロジェクトを作成した際 (86ページのSTEP3) に、[User Automatic Reference Counting] のチェックを外したかを確認しましょう。

Schemeでは、プログラムのビルドターゲットを設定します。XcodeはこのSchemeで指定したデバイス上でプログラムが動くようにコードをコンパイルし、対応したiOSシミュレータを起動します。今回はシミュレータ上で動作確認を行うため、[iPhone (バージョン番号) Simulator] か [iPad (バージョン番号) Simulator] と選択されていればOKです (87ページのSTEP5)。

また、[Cordova-based Application] テンプレートを使ってプロジェクトを作成した場合は、プロジェクト名とおなじライブラリ、Frameworks、Productsが組みこまれた状態で開発がスタートします (87ページのSTEP5)。これらはCordovaを使うにあたり、必須のライブラリファイルです。これらのファイルが誤って削除されていないか、確認してみましょう。

||||| アプリがうまく動作しない |||

次の3点を確認してみましょう。

- ・www/index.htmlが存在しているか

最初に、**A**のwindow.addEventListenerで、HTMLの読み込みが完了した際にinit関数を呼び出すように定義しています。addEventListenerについては後述します。

Bが、(1)の機能を担うinit関数、**C**が(2)の機能を担うcountUp関数です。

関数とは、一連の処理がひとまとまりになったものです。よく使用する処理や、複雑な処理を1つの関数にまとめることで、コードの視認性を高めたり、同じようなコードを書かずに済むメリットがあります。

関数はあらかじめ用意されているものもあれば、自分で作成することもできます。関数を自分で作成した場合、“関数を定義する”と呼びます。今回の2つの関数の場合、init関数とcountUp関数を定義した、ということになります。

また、関数には引数を渡すことができます。引数を受け取った関数は、渡された値に従って処理を行い、必要に応じて結果を返します。カンマで区切ることで、複数の引数を関数に渡すことができます。

|||| init関数を確認する - document.getElementById() |||

それでは(1)の機能を担う、init関数から見ていきましょう。init関数では、ボタンを押した時に(2)のcountUp関数を呼び出すように定義しています。

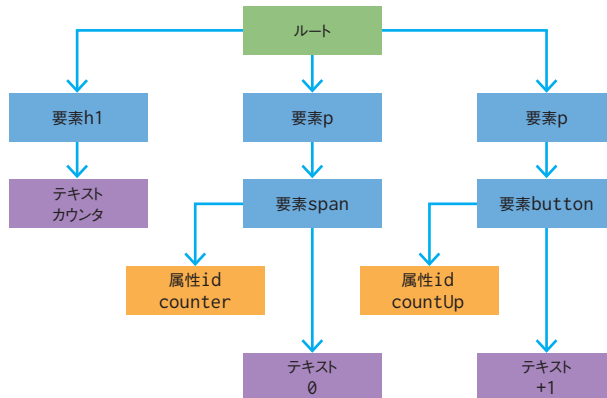
```
10 function init() {
11     document.getElementById( 'countUp' ).addEventListener( 'click' , countUp,
12     false);
12 }
```

ここではまず、document.getElementById()に注目してください。document.getElementById()は、IDを用いてDOMオブジェクトを得るメソッドです。

DOMの考え方を思い出してみてください。以下に<body>内のHTMLを抜粋します。

```
39 <h1>カウンタ</h1>
40 <p><span id="counter">0</span></p>
41 <p><button id=" countUp">+1</button></p>
```

WebブラウザがこのHTMLを読み込むと、次のようなDOMツリーを構築します。



document.getElementByIdメソッドに、IDを引数として渡すことで、DOMオブジェクトを得ることができます。ここでは引数に、「countUp」を渡しています。

```
11 document.getElementById('countUp')
```

この一文では、さきほどのDOMツリー図の中で、属性「id」ノードに「countUp」が格納されている要素ノードをオブジェクトとして取得しています。その結果、要素「button」ノードの情報を得ることができます。HTML中の場合、41行目にある<button id="countUp">+1</button>を指します。

```
41 <button id="countUp">+1</button>
```

||||| init関数を確認する - イベントリスナー |||||

続いて、document.getElementById('countUp')の後ろについている、.addEventListenerに注目してください。addEventListenerは、イベントリスナーを登録するメソッドです。

(133ページのColumn 03「用語について」も参照してください)

イベントとは、マウスのクリックや移動、キーの入力など、Webブラウザ上に発生した特定の操作を指します。

ユーザがWebブラウザ上で操作をしている間、裏ではさまざまなイベントが発生しています。代表的なものを列挙します。

- ・Webページがロードされた
- ・マウスが動いた
- ・マウスでクリックされた
- ・キーボードのキーが押された
- ・入力ボックスからフォーカスが外れた
- ・他のページに移動した

これらのイベントが発生した際に、対応する処理を指定しておけば、特定の操作が行われたあとに特定の処理が行われるようになります。

JavaScriptではこれらのイベントが発生した際にキャッチアップする仕組み(イベントハンドラ)が多数用意されています。先の例に対応するイベントハンドラは次のとおりです。

イベントハンドラの種類	
例	イベントハンドラ
Webページがロードされた	load
マウスが動いた	move
マウスでクリックされた	click
キーボードのキーが押された	keyDown、keyPress
入力ボックスからフォーカスが外れた	blur
他のページに移動した	unload

addEventListenerの「イベントリスナーを登録する」とは、これら特定のイベントが発生した際に、任意の処理が行われるようにする意味となります。

addEventListenerには、3つの引数を渡します(*)。

| スクリプト | addEventListenerの書式

```
addEventListener(type, listener, useCapture)
```

(※)書式は、JavaScriptの中核的な仕様を策定しているECMAScriptより。

addEventListenerの引数		
引数	内容	型
type	イベントのタイプを指定します	文字列
listener	typeで指定したイベントが発生したときに呼び出す関数を指定します	関数
useCapture	イベントの伝搬方向をtrueかfalseで指定します。trueを指定した場合、ドキュメントルートからイベント発生源に順にイベントが渡っていきます (CAPTURING_PHASE)。falseを指定した場合、イベント発生源からドキュメントルートまで順にイベントが渡っていきます (BUBBLING_PHASE)	論理値

複数のボタンや要素に、複雑なイベントリスナーを設定した場合、イベントリスナーが干渉し合い、同じ処理が複数回発生してしまう場合があります。このような場合に3番目の引数にてイベントの伝搬方向を指定することで、意図する動作にすることが可能です。本書のサンプルでは、一貫して「false」を指定します。

```
01 addEventListener('click', countUp, false)
```

上のコードは、マウスがクリックされた際に、(2)のcountUp関数を呼び出すように、という意味になります。

