

# スマートデバイス : Monacaを 使ったモバイルアプリケーション 開発の基礎

アシアル株式会社  
生形 可奈子



# 自己紹介



生形 可奈子（うぶかた かなこ）

アシアル株式会社

Monaca / Onsen UI エバンジェリスト  
モナカプレス 編集長

セミナー講師・書籍執筆・オウンドメディア運営など、モバイルアプリ開発技術の普及・促進を目的とした活動を行っています。

モナカプレス

<https://press.monaca.io>



# 本日の内容

---

1. ハイブリッドアプリの概要
2. Monacaの紹介・セットアップ
3. HelloWorldアプリの実行
4. Webサービスとの連携
5. サンプルアプリの作成
6. アプリのビルド・配布について
7. Proプランへのアップグレード方法

# ハイブリッドアプリの概要

---



# モバイルアプリ開発に関する課題

---



×



Objective-C

開発言語がOS毎に異なるため、

1. 開発工数がかかる
2. ソースコード管理の複雑化
3. エンジニアの確保が困難



×



# ハイブリッドアプリの登場

ネイティブアプリとWebアプリ、2つのアプリの特徴を兼ね備えたアプリを「ハイブリッドアプリ」と呼びます。

ネイティブアプリ



Webアプリ

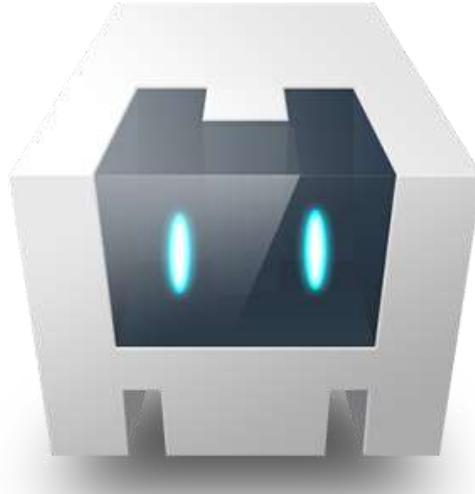
ハイブリッドアプリはWebの技術を使って開発しますが、生成されるアプリはネイティブアプリと同等のものになります。

# 従来の開発手法とハイブリッドアプリの比較

特徴・性能	ネイティブアプリ	Webアプリ	ハイブリッドアプリ
クロスプラットフォーム対応	×	○	○
端末へのインストール	○	×	○
マーケットでの配布	○	×	○
オフラインでの利用	○	×	○
端末固有の機能の利用	○	△	○
アプリ実行速度	○	△	△

# HTML5モバイルアプリ用フレームワーク

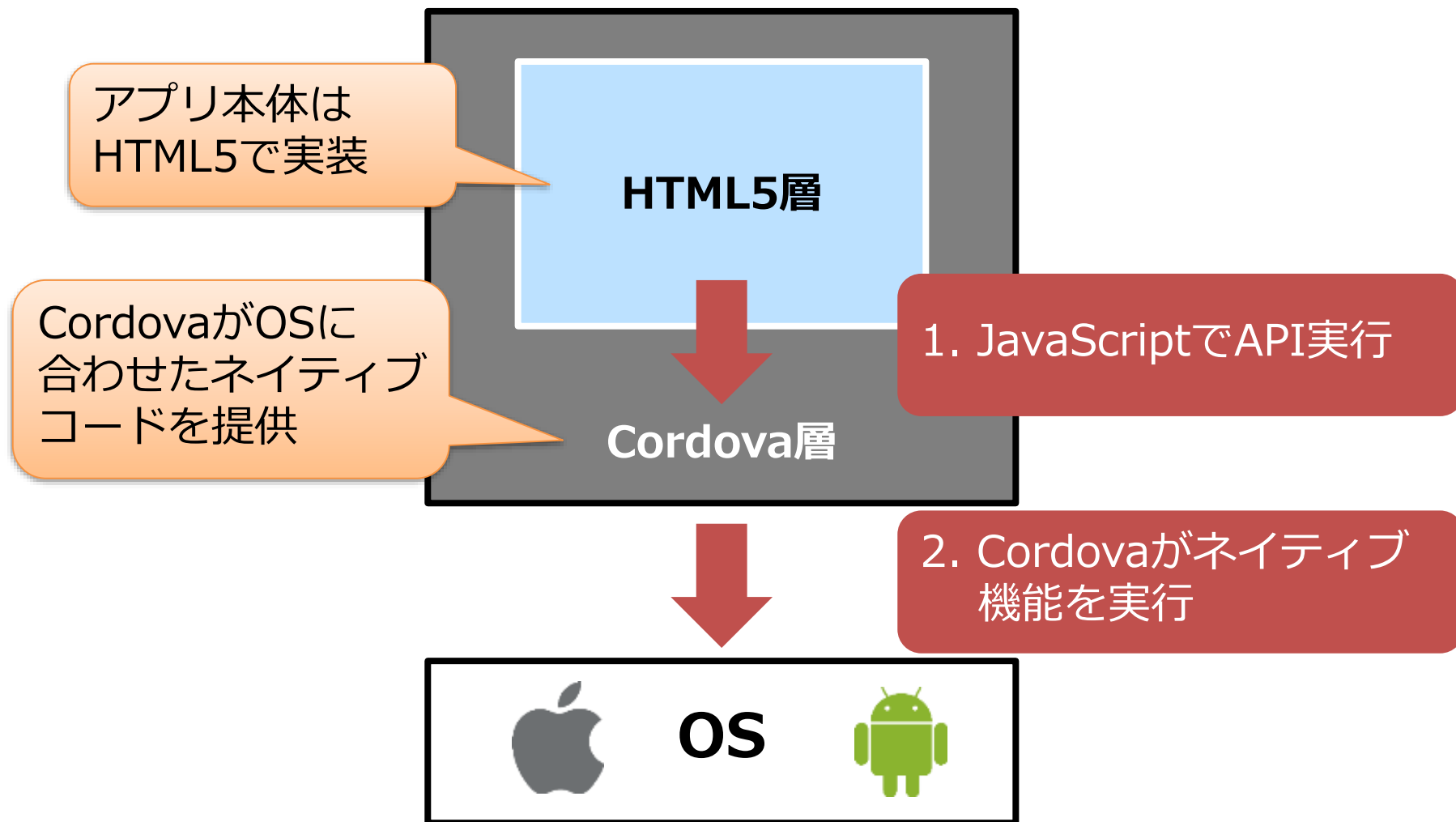
---



Cordova (旧PhoneGap)  
Apacheソフトウェア財団



# Cordovaの仕組み



# Monacaの紹介

---



# Cordova開発環境 : Monaca

Monacaは日本でもっとも普及しているCordova開発環境の一つです。



18万人が利用する  
Cordova開発環境

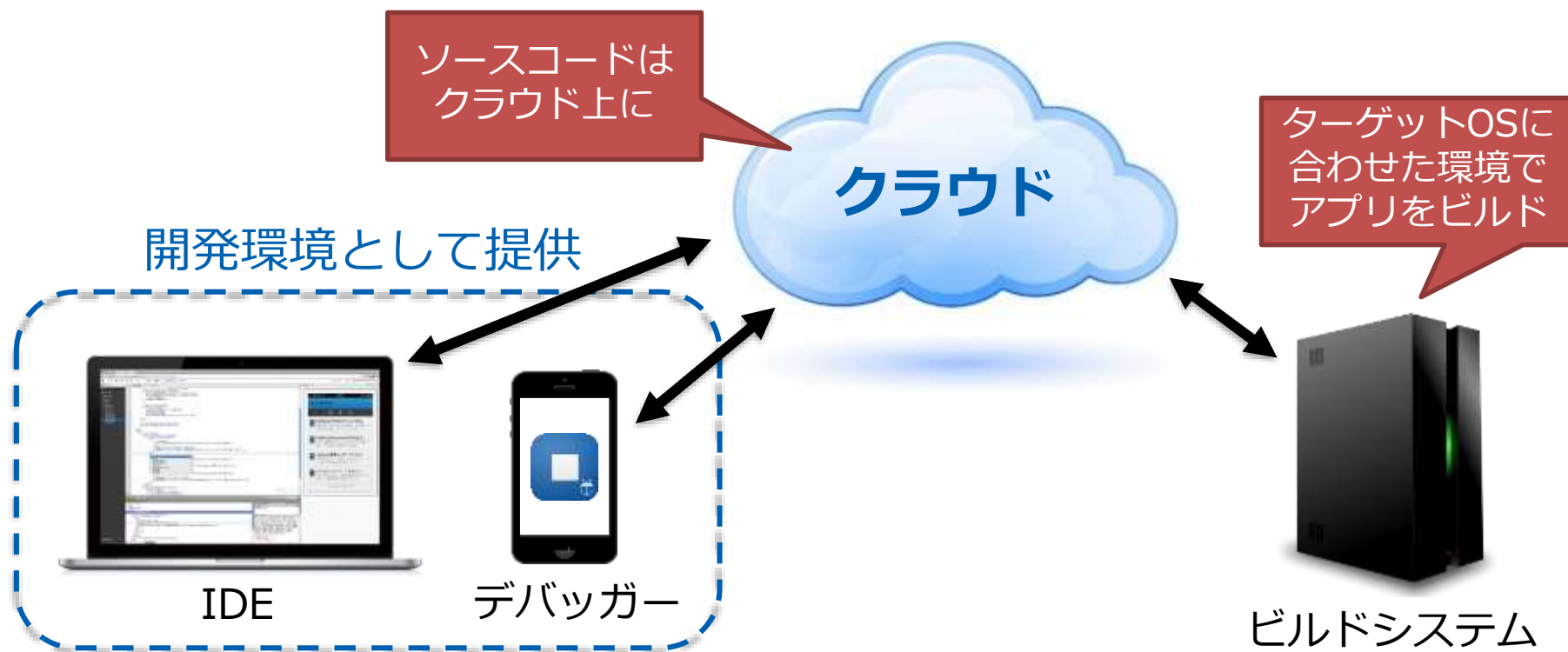
実機でのリアルタイム  
検証が可能

UIフレームワーク  
Onsen UI搭載

安心の日本語サポート

# Monacaの仕組み

IDE、ビルド環境はクラウドサービスとして提供。  
どんな環境でもアプリ開発を始められます。



# 選べるIDE

開発スタイルに合わせて様々なIDEを選択することが可能です。

## Cloud IDE

ブラウザーベースの  
フル機能IDE



## Monaca Localkit

既存の開発環境が  
そのまま利用可能



## Monaca CLI

コマンドライン  
からの実行



**Jenkins**

**Monaca Proプラン以上で利用可能**

# Monacaデバッガー

コンパイル処理やUSB経由での実機転送などは不要。  
デバッグ専用アプリがネットワーク経由で変更箇所を  
取得するため、リアルタイムに動作検証できます。

①ファイルを編集

②実機ですぐに動作確認



推奨環境  
Google Chrome



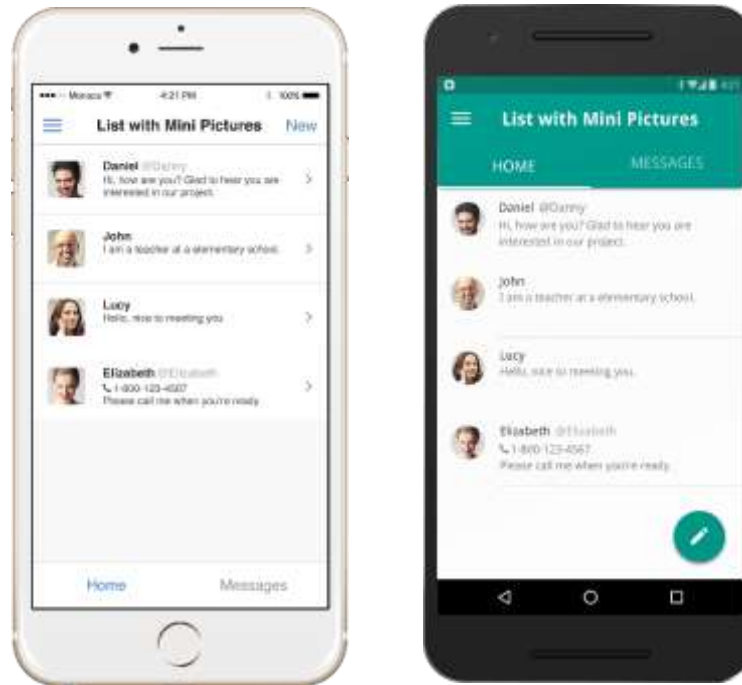
# HTML5モバイルアプリ専用 UIフレームワーク

---



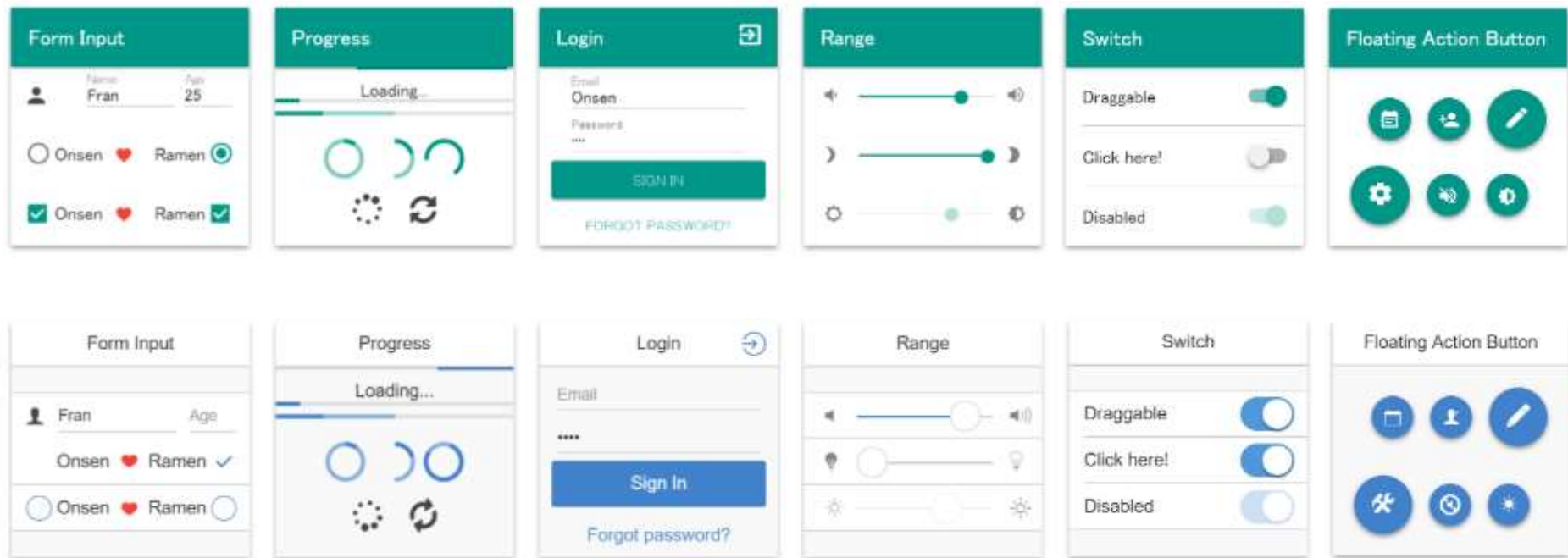
# Onsen UI

- ハイパフォーマンスなUIを実現
- プラットフォームを判別して自動でスタイルが変化





# 豊富なUIパーツを利用可能



# 使い方は独自タグを記述するだけの 簡単設計



→ `<ons-toolbar></ons-toolbar>`

→ `<ons-list-item></ons-list-item>`

→ `<ons-tabbar></ons-tabbar>`

# 対応JavaScriptフレームワーク

- Onsen UIはJavaScriptフレームワークに依存しないため、任意のフレームワークと組み合わせて利用することができます。
- もちろん、フレームワークなしで利用することも可能です。



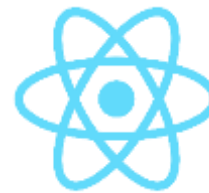
jQuery



AngularJS



Angular2+



React



Vue.js

# Monacaのセットアップ

---



# Monacaのアカウント登録

ブラウザでMonacaのWebサイトを開き、「サインアップ」をクリックしてアカウント登録して下さい。

<http://ja.monaca.io/>



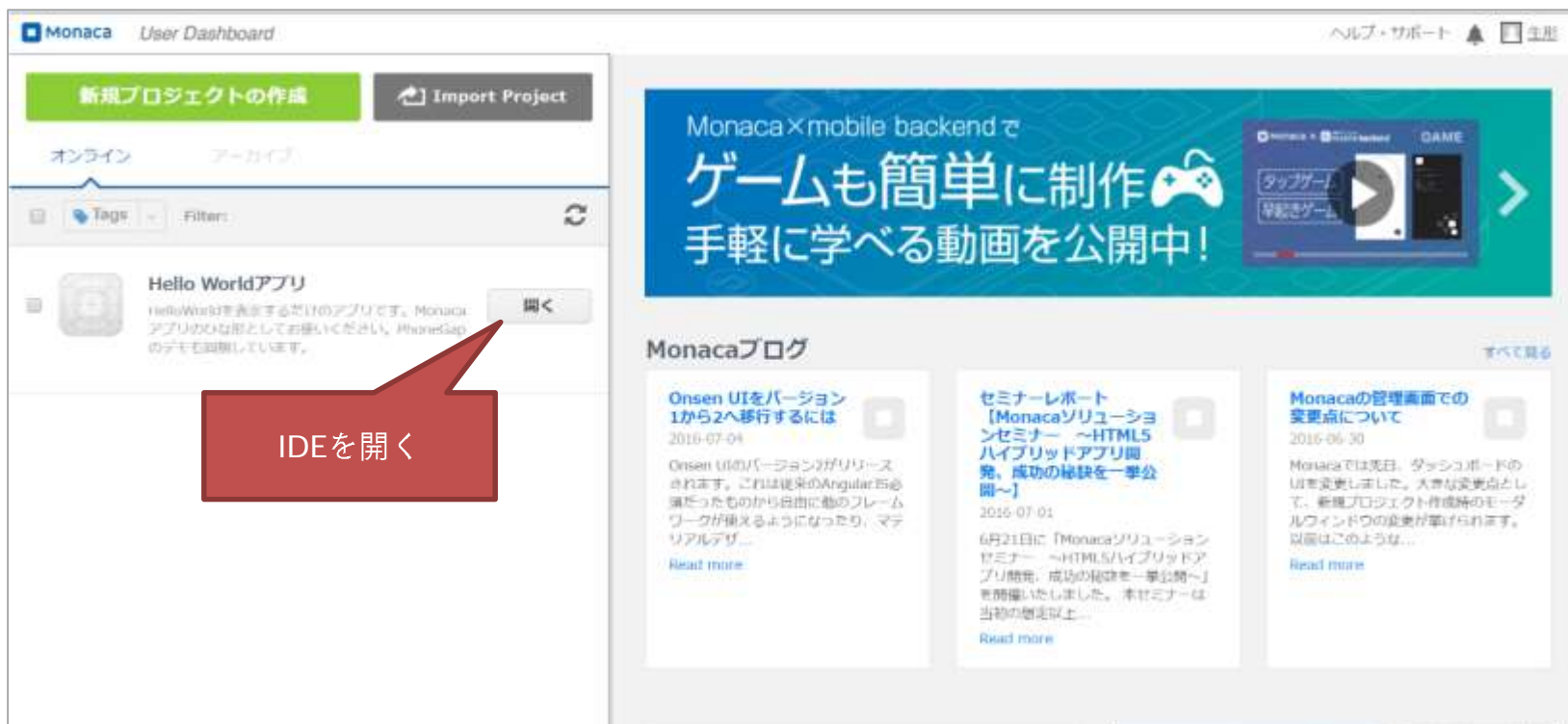
# アカウント仮登録完了

登録したメールアドレスに確認のメールが届きます。メールに記載されたURLにアクセスすることで登録完了です。



# IDEを起動する

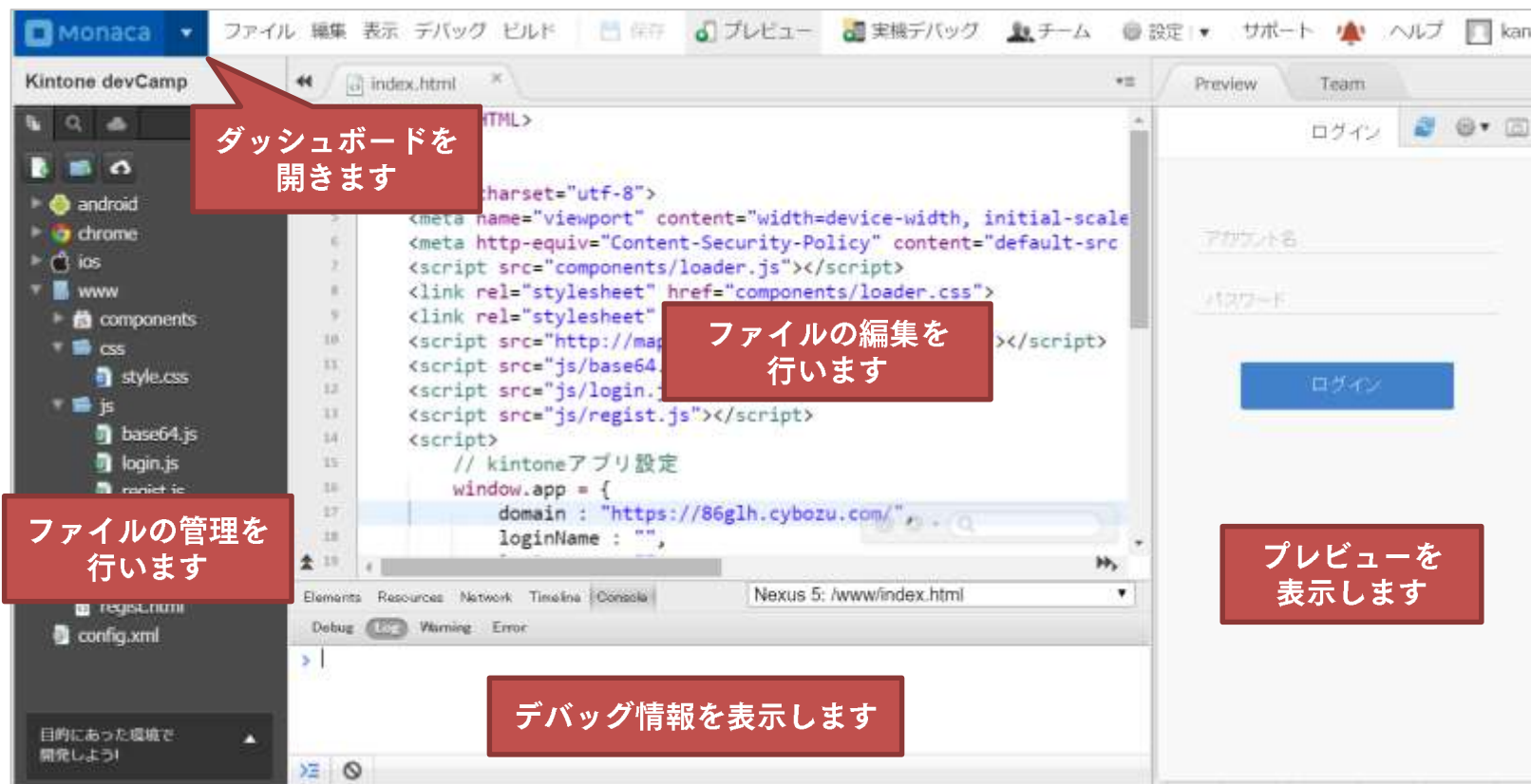
プロジェクトの「開く」ボタンをクリックすると、IDEが起動します。





# IDEの起動

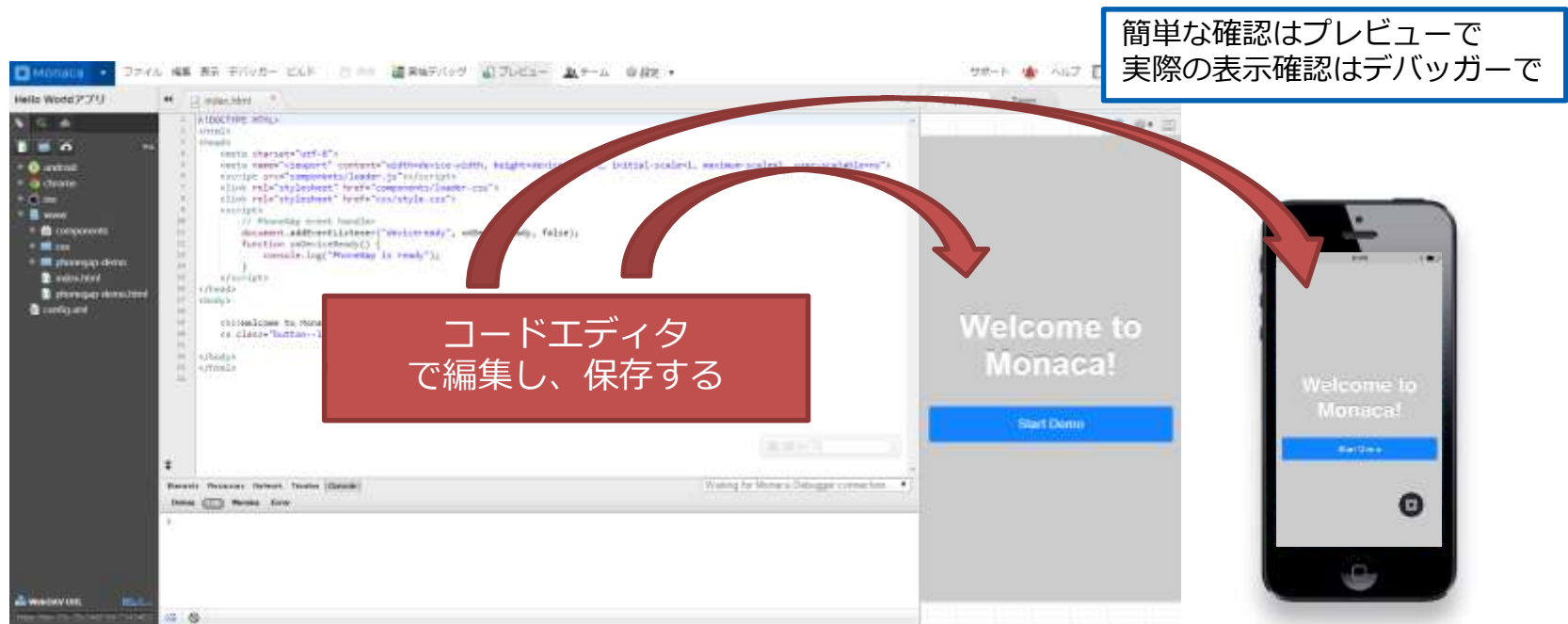
IDEの各部の役割は以下の通りです。





# 開発の流れ

コードエディタで編集したファイルを、プレビューとデバッガーで確認しながら開発を進めていきます。



# Monacaデバッガーのインストール

App Store、またはGoogle Playにて「monaca」で検索し、スマートフォンにインストールして下さい。



アイコンはこちらです。

# Monacaデバッガーの起動



Monacaデバッガーを起動すると、左のログイン画面が表示されます。

Monacaに登録したアカウントでログインを行ってください。

# Monacaデバッガーでアプリを実行する

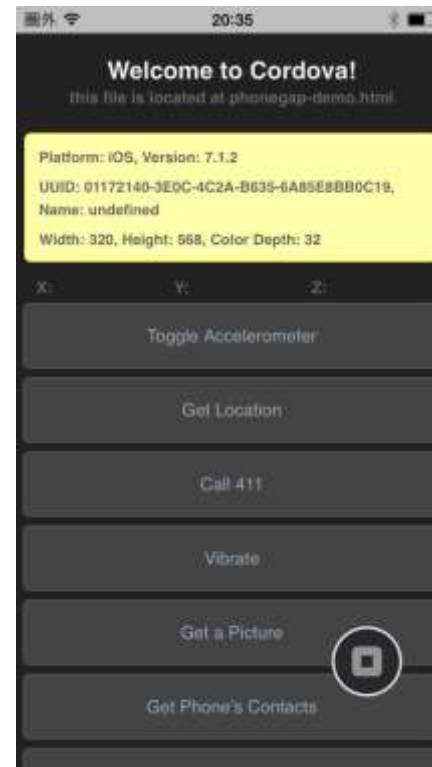
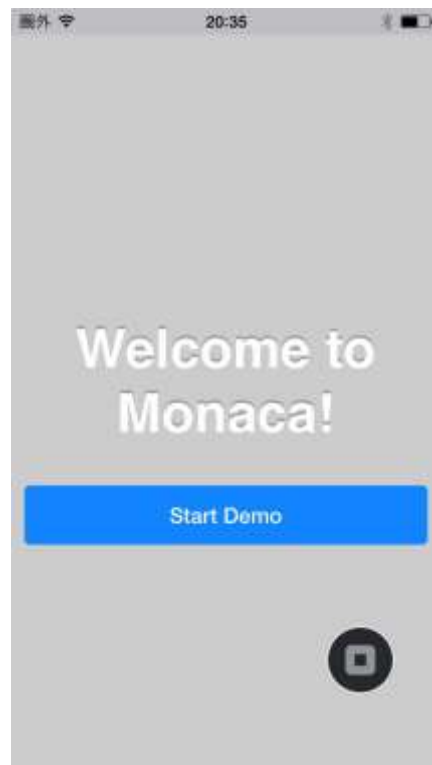
デバッガーを起動すると自動的にクラウドへ接続し、プロジェクトが一覧表示されます。

プロジェクトをタップするとアプリがシミュレートされます。

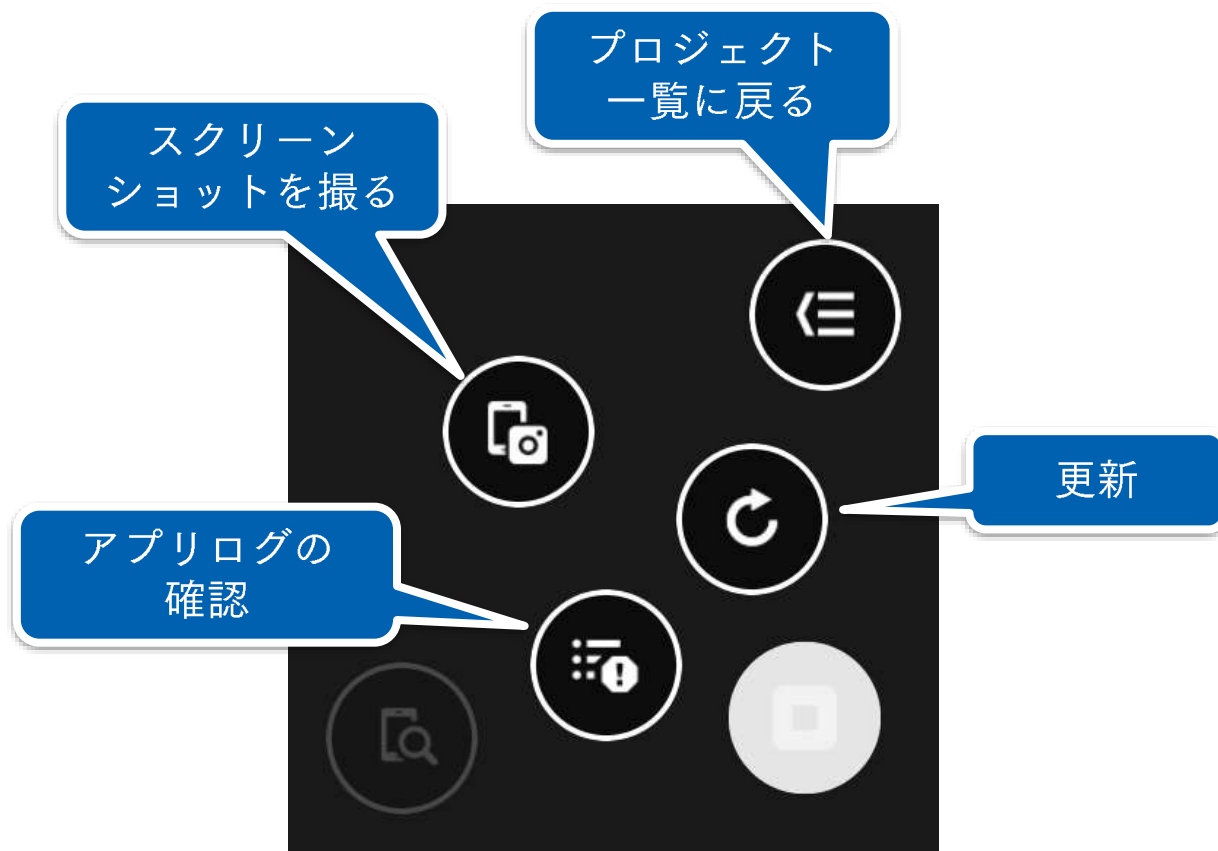


# 「Hello Worldアプリ」を動かしてみる

カメラ、コンパス、バイブレーションなどのネイティブ機能を動かすことができます。



# Monacaデバッガーのメニュー



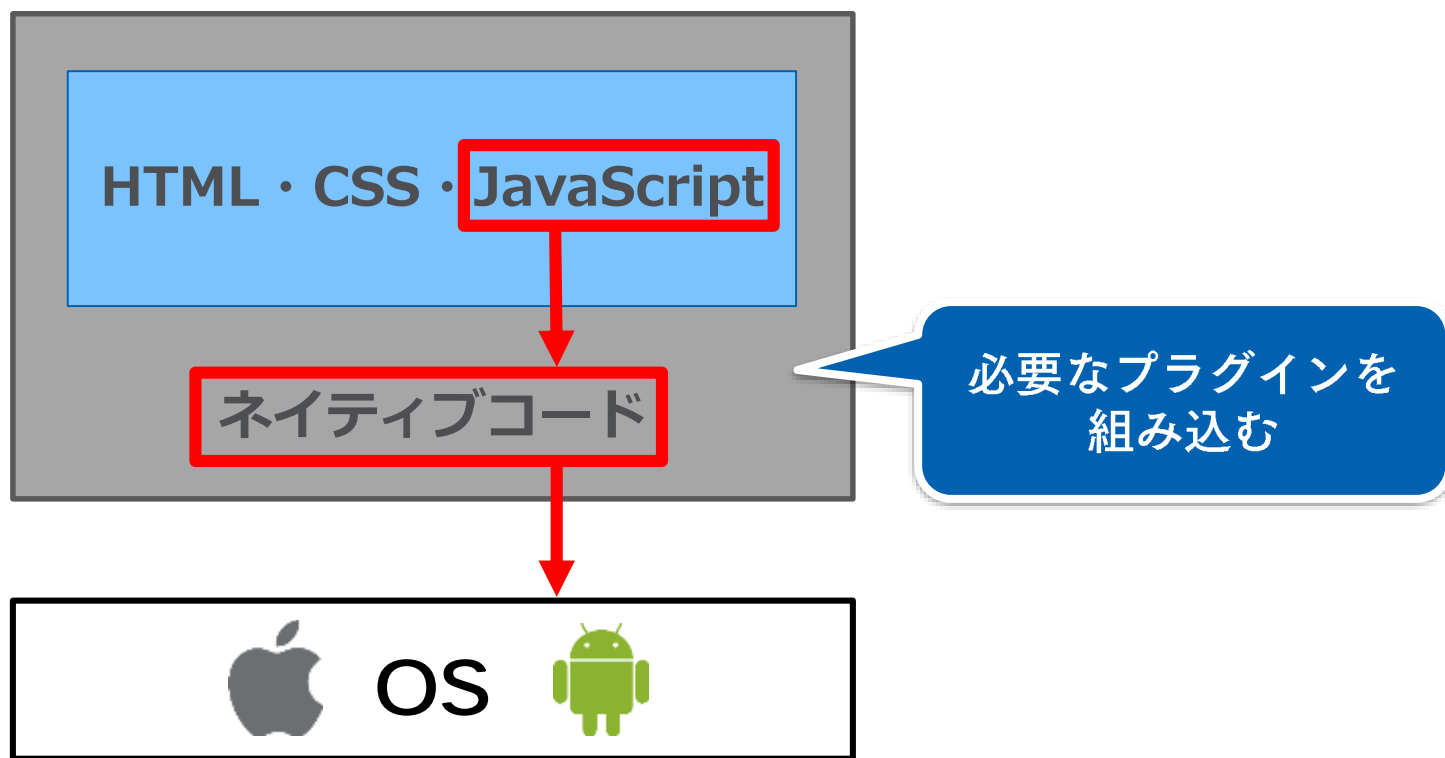
# Cordovaプラグインについて

---



# ネイティブ機能呼び出しの仕組み

ネイティブの各機能は、機能ごとにプラグイン形式で提供されています。開発者はJavaScriptでAPIをコールすることで、プラグインが実行され端末固有の機能にアクセスすることができます。





# Cordovaプラグインの種類

---

## ■ 標準プラグイン

- カメラ、位置情報、ファイルアクセス等の基本的な機能を提供
- Monacaデバッガーにあらかじめ組み込まれている

## ■ 第三者提供のプラグイン

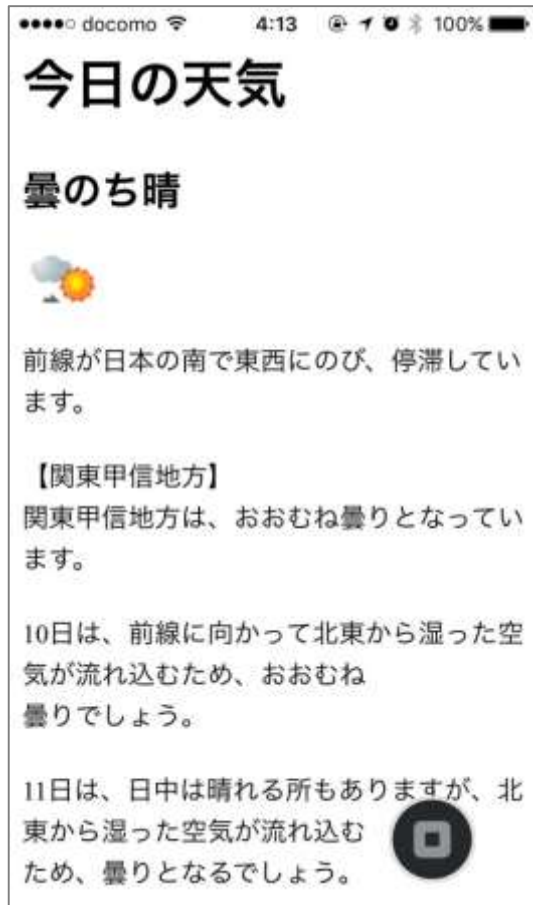
- <https://cordova.apache.org/plugins/>
  - 2000を超えるプラグインが公開済
- ネイティブ言語を用いた自作プラグインの作成も可能
- アプリの動作検証をするにはビルドが必要
- Proプラン以上で利用可能

# 今回作成するアプリ

---



# お天気情報閲覧アプリ



Web APIを用いて東京の天報情報を取得し、画面に表示します。

Monacaによるアプリ開発手法と、外部のWebサービスを連携する方法を学びます。

# 外部Webサービスの利用

---



# 外部サービスとの連携方法

---

以下の3つのうち、いずれかが提供されていればMonacaからの利用が可能です。

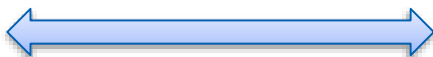
- Cordovaプラグイン
- JavaScript SDK
  - ・ クライアントサイド用に限る
  - ・ プロジェクトへのJSファイルのアップロード、またはCDNからの読み込みのどちらでも可
- Web API

# Web API連携で機能拡張

Web API連携することでプッシュ通知、IoT、人工知能などを絡めたアプリも開発可能となります。



JavaScriptによる  
データの送受信



## 各種クラウドサービス

- ・プッシュ通知
- ・クラウドデータベース
- ・ユーザー認証
- ・人工知能
- ・IoT連携



IBM Watson

NIFTY Cloud

mobile backend



Microsoft Azure

# API提供サイト : Weather Hacks

livedoorが運営する、天気情報をWeb APIで提供するWebサービスです。

[http://weather.livedoor.com/weather\\_hacks/webservice](http://weather.livedoor.com/weather_hacks/webservice)



# JavaScriptによる非同期通信

---

## ■ XMLHttpRequest (XHR)

古くから存在するHTMLのAPIです。どのブラウザでもサポートされています。記述方法が冗長になるため、そのまま使うことはほぼありません。

## ■ JavaScriptフレームワークを使う

jQueryやAngularなどでは、XHRをラップしたメソッドを提供しています。

## ■ HTML5 の Fetch API を使う

HTML5から追加された、標準的な非同期通信用のAPIです。現在では多くのブラウザでサポートされています。



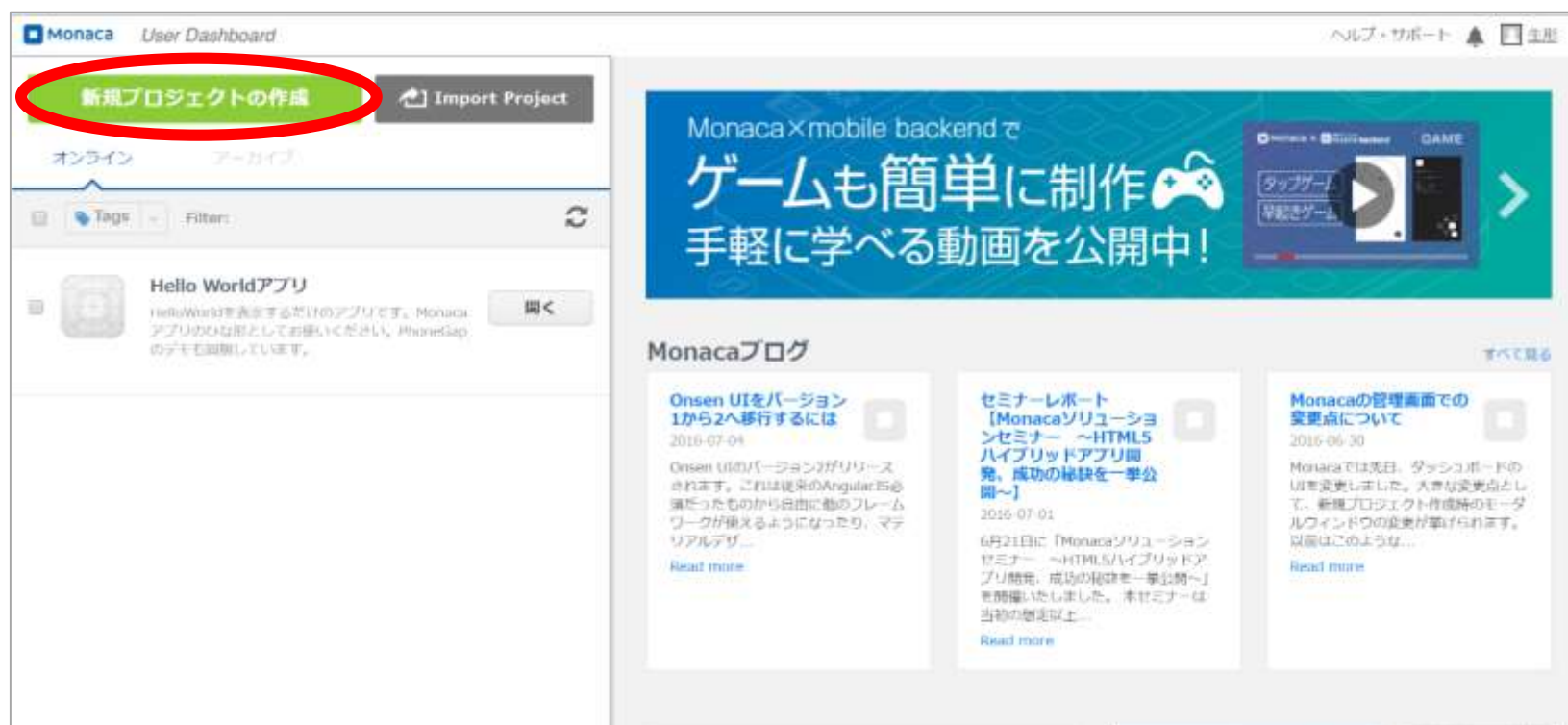
# アプリの作成

---



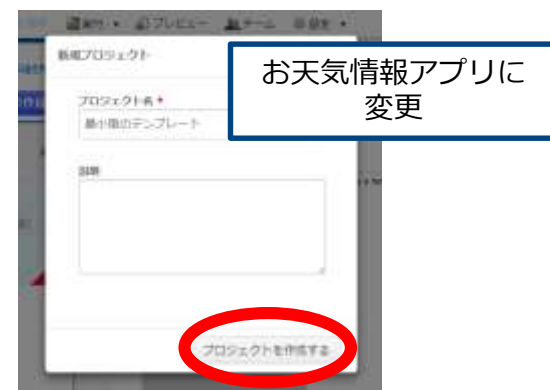
# 新しいプロジェクトを作成する

Monacaにログインし、ダッシュボードで「新規プロジェクトの作成」ボタンを選択します。



# 新しいプロジェクトを作成する

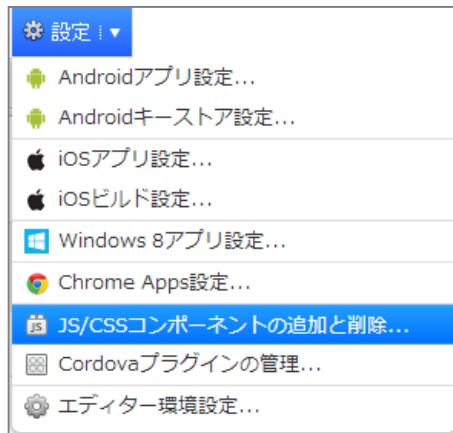
1. プロジェクトのひな型から「No Framework」「最小限のテンプレート」を選択します
2. 名前と説明（任意入力）を設定して新規プロジェクトを作成します
  - プロジェクト名：**お天気情報アプリ**
3. プロジェクトの作成が完了したら、ダッシュボードの「開く」ボタンを選択します



# jQueryの有効化

「設定」メニューから「JS/CSSコンポーネントの追加と削除...」を選択し、「jQuery (Monaca Version)」の「追加」ボタンをクリックしてください。

「インストール開始」ボタンを押して、以降はすべて青いボタンをクリックします。



# HTMLの実装

index.htmlの<body>タグを以下のように書き換えます。

```
<body>
  <div>
    <h1>今日の天気</h1>
    <h2 id="label"></h2>
    <img id="image" src="">
  </div>
  <p id="description"></p>
</body>
```

# スクリプトの実装

完成したら、Monacaデバッガーで実行してみましょう。

index.htmlの<script>タグ内に以下のコードを追記します。

```
$.ajax({
  url: 'http://weather.livedoor.com/forecast/webservice/json/v1',
  data: {
    city: '130010' // 東京のコード
  }
})
.done((result) => {
  $('#label').text(result.forecasts[0].telop);
  $('#image').attr('src', result.forecasts[0].image.url);
  $('#description')
    .html(result.description.text.replace(/¥n/g, '<br>'));
})
.fail((error) => {
  alert('エラーが発生しました');
});
```

# \$.ajax()の利用

```
$.ajax({キーと値のセット}).done(成功時処理).fail(失敗時処理);
```

キー	値
url	リクエストの送信先
dataType	取得するデータの種類 (xml/html/jsonなど) ※省略時は自動判別される
method	HTTPメソッド (GET/POST/PUTなど) ※省略時はGET
timeout	通信がタイムアウトする時間をミリ秒単位で設定
data	リクエストパラメータを指定

# アプリのビルド・配布

---





# アプリ設定

メニューバー「設定」 > 「iOS/Androidアプリ設定」画面で、アプリ名やパッケージ名、アイコン画像等を設定します。

ビルド/ビルド設定  
Android  
iOS  
Windows

アプリの設定  
Android  
iOS  
Windows

プロジェクト  
Carthageプラグイン  
X/CSSコンポーネント  
サービス連携

ビルド  
最終的インテグレーション

サービス  
デプロイサービス

### Androidアプリ設定

#### アプリケーション情報

アプリケーション名: J2EE HelloApp

パッケージ名: com.example.monaca

ビルド種別ごとにパッケージ名を分ける: ☒

バージョン: 1.0.0

バージョンコード: ☐ バージョンコードを指定

フルスクリーン: ☒

#### アイコン

PNG形式の画像を指定してください。

一括で設定する  
アップロード

保存する

# ビルドの種類

Android版		iOS版
デバッグビルド	主にテスト用途で利用します。アプリを社内限定で配布する場合などにも利用可能です。ダミーの署名をつけてビルドします。	テスト用途で開発者向けにアプリを配布する場合に利用します。Developer証明書が必要です。
アドホックビルド	—	特定の端末にアプリを限定配布する場合に利用します。ディストリビューション証明書とAdHoc用のProvisioningプロファイルが必要です。
In-Houseビルド	—	組織内にアプリを配布する場合に利用します。ディストリビューション証明書とIn-Houseビルド用のProvisioningプロファイルが必要です。
リリースビルド	Google Playで公開することができます。自分で作成した署名をつけてビルドを行います。	App Storeで公開することができます。ディストリビューション証明書とApp Store用のProvisioningプロファイルが必要です。
カスタムビルドデバッガー	第三者提供のCordovaプラグインを含むカスタムデバッガーを作成します。	

# 【参考】Androidデバッグビルド (1/2)

Androidビルド画面を開くと[デバッグビルド]が選択されているので、そのまま[ビルドを開始する]ボタンをクリックします。



# 【参考】Androidデバッグビルド (2/2)

ビルドが開始されます。ビルドが完了するまでには数分の時間がかかります。



# スマートフォンにインストールする

QRコードから直接端末にインストールできます。

リリースビルドの場合、ローカルPCにダウンロードしたapkファイル（iOSの場合はipaファイル）はそのままストアに公開可能です。

## ビルド



Android リリースビルド

Project: v1.0.0

🕒 数秒前

📅 2017年2月16日 17:12:17

おめでとうございます！

ビルドに成功しました。ダウンロードしてデバイスにインストールしてください。ビルドログを確認するにはこちらをクリックしてください。



ローカルPCに  
ダウンロード



QRコードから  
インストール



登録メールアドレスに  
インストール方法を通知

 Monaca

URL : <http://ja.monaca.io/>

Copyright © Asial Corporation. All Right Reserved.

53

# 【iOS】ビルド設定とリリース向けビルド

---

iOSアプリのビルドには、有償のApple Developer Programへの参加と、証明書の発行が必要になります。

以下のドキュメントを参考に作業を行って下さい。

**[https://docs.monaca.io/ja/monaca\\_ide/manual/build/ios/](https://docs.monaca.io/ja/monaca_ide/manual/build/ios/)**

# Proプランへのアップグレード方法

---



# プランのアップグレード

①ダッシュボード下部の  
[アクティベーションコード]  
をクリック



②ユーザー情報を入力

ユーザー情報を入力してください

氏名 \*

所属 (学校・会社)

電話番号 \*

③12桁のアクティベーションコード  
(ハイフンあり) を入力して適用

アクティベーションコードの利用

アクティベーションコードを入力してください。一度に3個までのアクティベーションコードを入力できます。



# 参考ドキュメント

---



# 参考ドキュメント

---

- 標準Cordovaプラグインリファレンス

**[https://docs.monaca.io/ja/reference/cordova\\_6.5/](https://docs.monaca.io/ja/reference/cordova_6.5/)**

- クラウドIDEマニュアル（ビルドやアプリの配布について）

**[https://docs.monaca.io/ja/monaca\\_ide/manual/](https://docs.monaca.io/ja/monaca_ide/manual/)**

- teratail（質問フォーラム）

**<https://teratail.com/tags/monaca>**

- モナカプレス（サンプルアプリの紹介やリリース情報など）

**<https://press.monaca.io/>**



<https://ja.monaca.io/>