

Monacaによる HTML5モバイルアプリ開発入門

アシアル株式会社
生形 可奈子



<https://bit.ly/2LNGvsc>



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

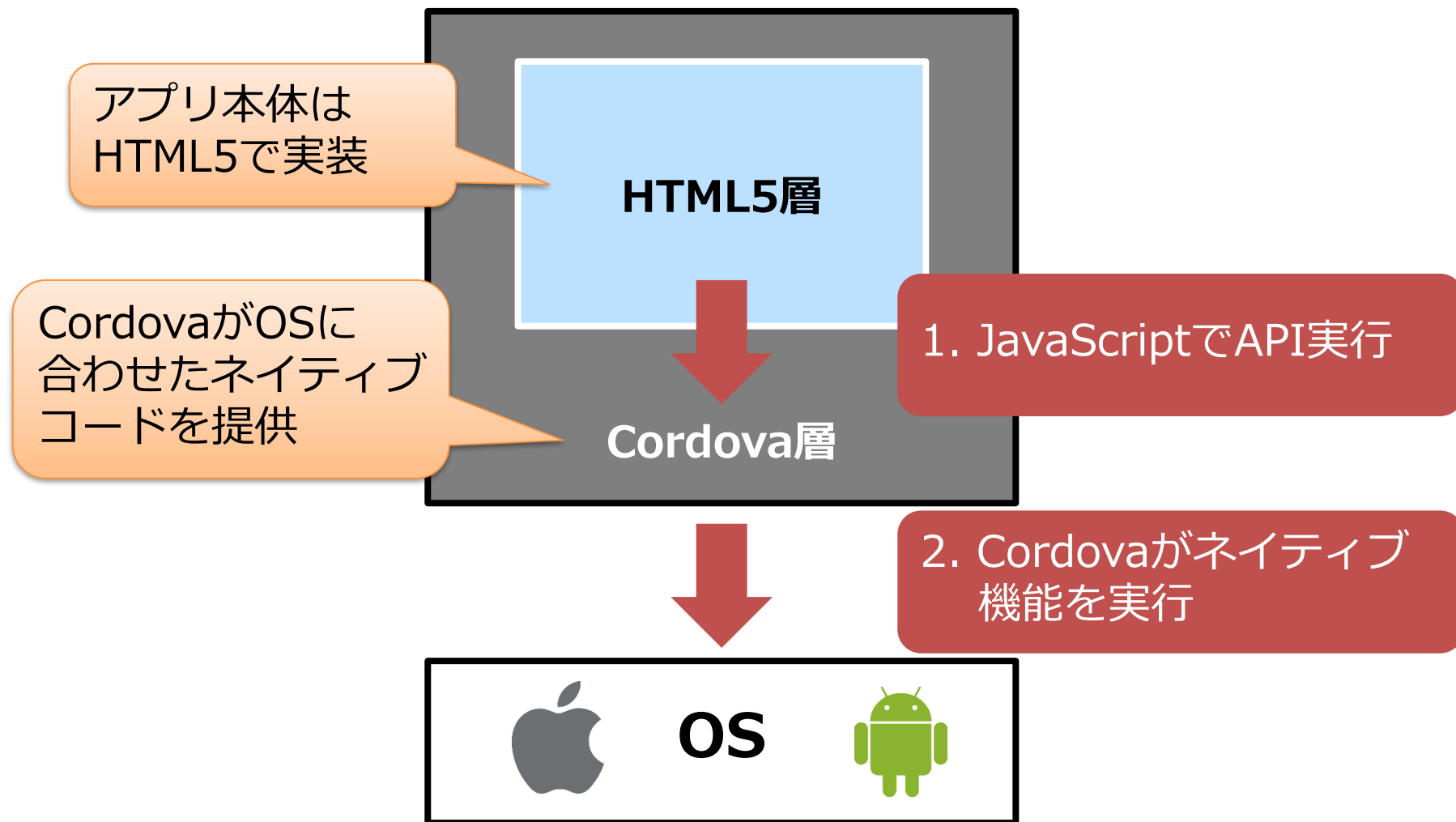
特徴・性能	ネイティブアプリ	Webアプリ	ハイブリッドアプリ
クロスプラットフォーム対応	×	○	○
ストアへの公開	○	×	○
端末へのインストール	○	△	○
オフラインでの利用	○	△	○
端末固有の機能の利用	○	△	○
アプリ実行速度	○	△	△

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。



20万人が利用する
Cordova開発環境

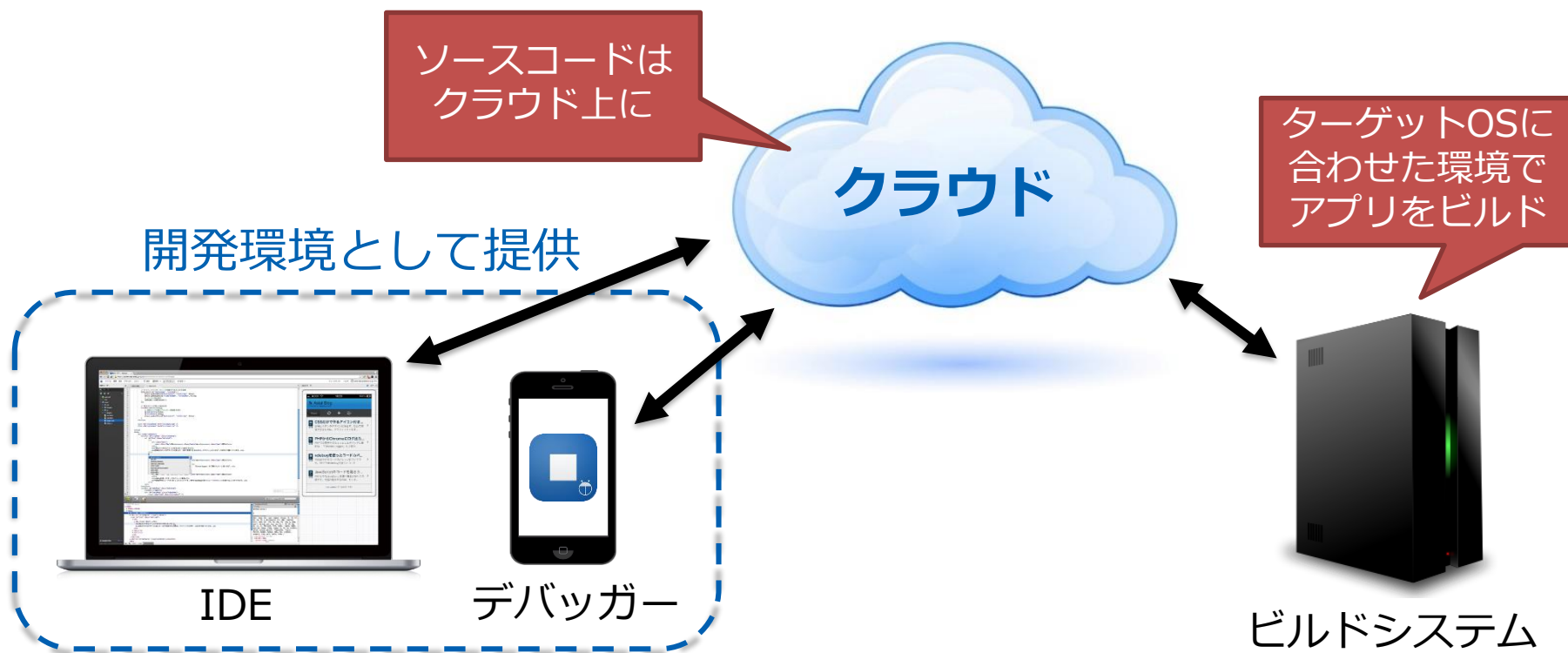
実機でのリアルタイム
検証が可能

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



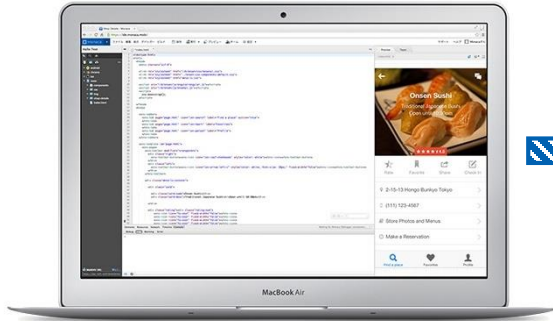
Jenkins

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

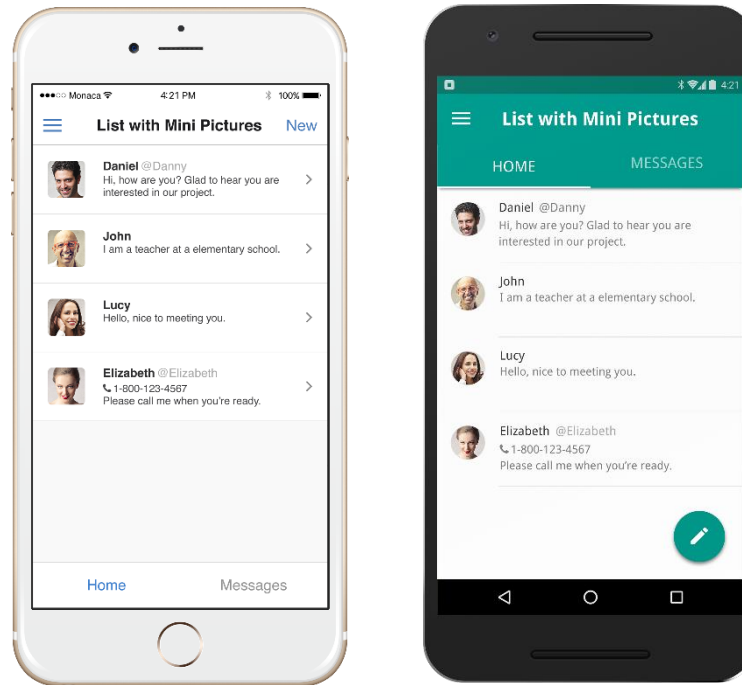


HTML5モバイルアプリ専用 UIフレームワーク

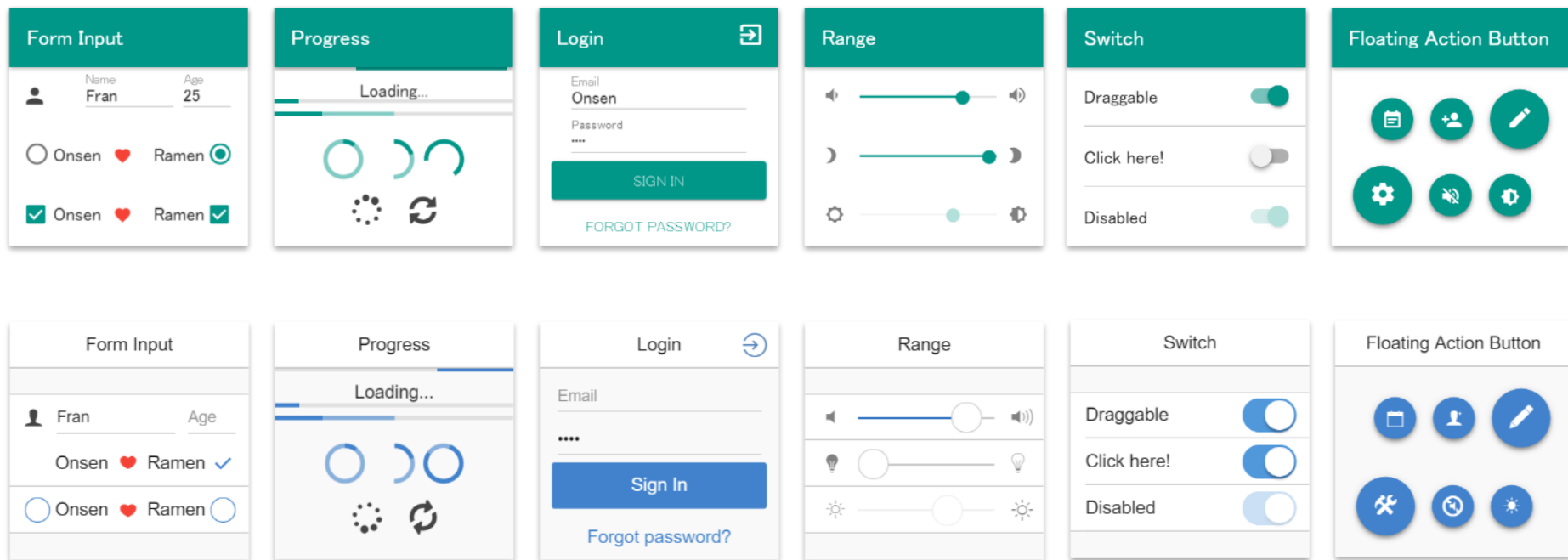


Onsen UI

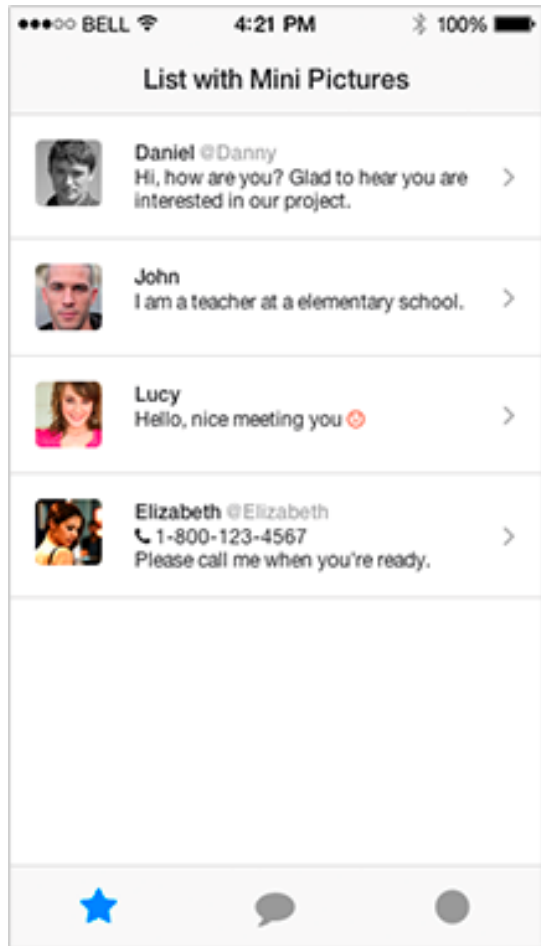
- ハイパフォーマンスなUIを実現
- プラットフォームを判別して自動でスタイルが変化



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。



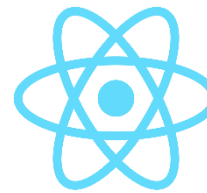
jQuery



AngularJS



Angular2+



React



Vue.js

Monacaのセットアップ



Monacaのアカウント登録

Chromeブラウザで下記URLを開き、「無料トライアル」ボタンをクリックしてアカウント登録を行って下さい。

<https://ja.monaca.io>

A promotional banner for Monaca. It features a hand holding a smartphone displaying a mobile app interface with a red and yellow geometric pattern and the text 'Kyoto, Japan & Bangkok'. In the background, a laptop screen shows the Monaca web interface with various navigation tabs like '特徴', '実績', '料金プラン', '導入支援', and '開発リソース'. The Monaca logo is in the top left corner. Overlaid text in the center reads 'あらゆるニーズに応える No.1アプリ開発クラウド' and 'Monacaは20万人以上のWeb技術者に選ばれたモバイルアプリ開発環境です'. An orange button with the text '無料トライアル' is prominently displayed, and a link 'アプリ開発のご相談はこちら' is at the bottom.

Monaca

あらゆるニーズに応える
No.1アプリ開発クラウド

Monacaは20万人以上のWeb技術者に選ばれたモバイルアプリ開発環境です

無料トライアル

アプリ開発のご相談はこちら ①

Monacaのアカウント登録

受信可能なメールアドレスとパスワードを入力してアカウント新規作成を行って下さい。

アカウント作成

メールアドレス **【必須】**

パスワード **【必須】**

半角英字と数字を組み合わせた7文字以上

アカウント作成ボタンをクリックすると、[利用規約](#)に同意したとみなされます。

アカウント新規作成

or

GitHubアカウントで作成

既にアカウントをお持ちですか？ [ログイン](#)

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にアクセスして、無料トライアルを開始して下さい。



IDEを起動する

プロジェクトを選択し、右側に表示される「クラウドIDEで開く」ボタンをクリックすると、IDEが起動します。



The screenshot shows the Monaca Dashboard interface. On the left, there's a sidebar with a '新しいプロジェクトを作る' (Create new project) button and an 'インポート' (Import) button. Below these, there's a list of projects, with 'HelloWorld' selected. The main area on the right shows the 'HelloWorld' project details, including a '開発' (Develop) tab and a 'クラウド開発' (Cloud development) section. A red speech bubble points to the 'クラウドIDEで開く' (Open with Cloud IDE) button, with the text 'IDEを開く' (Open IDE) inside it.

Monaca Dashboard

新しいプロジェクトを作る インポート

タグ オンライン / アーカイブ

HelloWorld
プロジェクトの概要がありません。

NEW

HelloWorld

プロジェクトの概要がありません

作成日: 2018-6-1 / 更新日: 2018-6-1 / フレームワーク: Cordova 7.1.0

開発 ビルド デプロイ 設定

クラウド開発

MonacaクラウドIDEはブラウザだけでご利用いただける開発環境です。コーディング、デバッグ、ビルドといった開発に必要なすべての機能が備わっています。

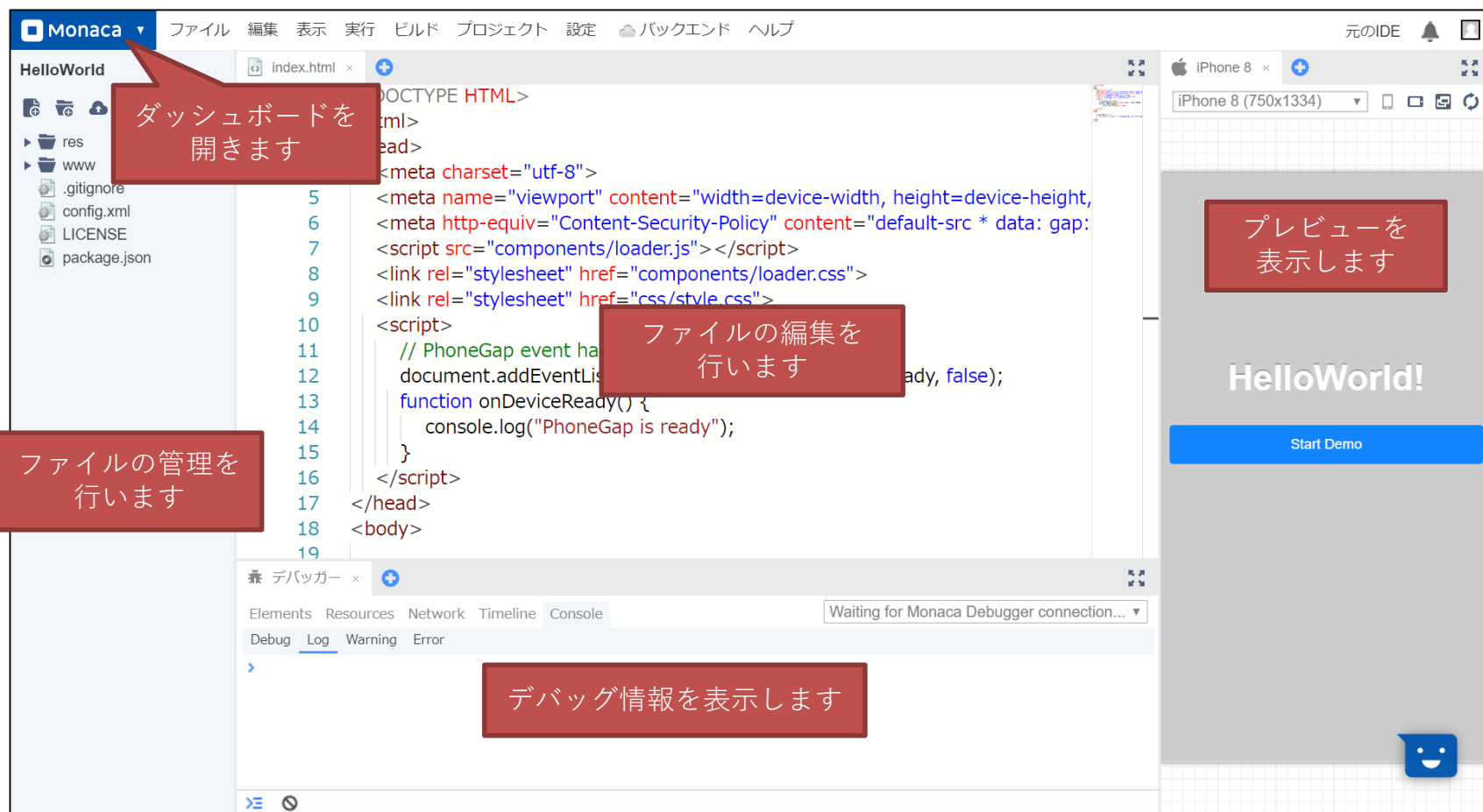
クラウドIDEで開く

コマンドラインツールを使う（ローカル開発）

あなたのコンピュータ上で好きなエディタやIDEと組み合わせて開発をするにはMonaca CLIをお使いください。詳しくは[マニュアル](#)をご覧ください。

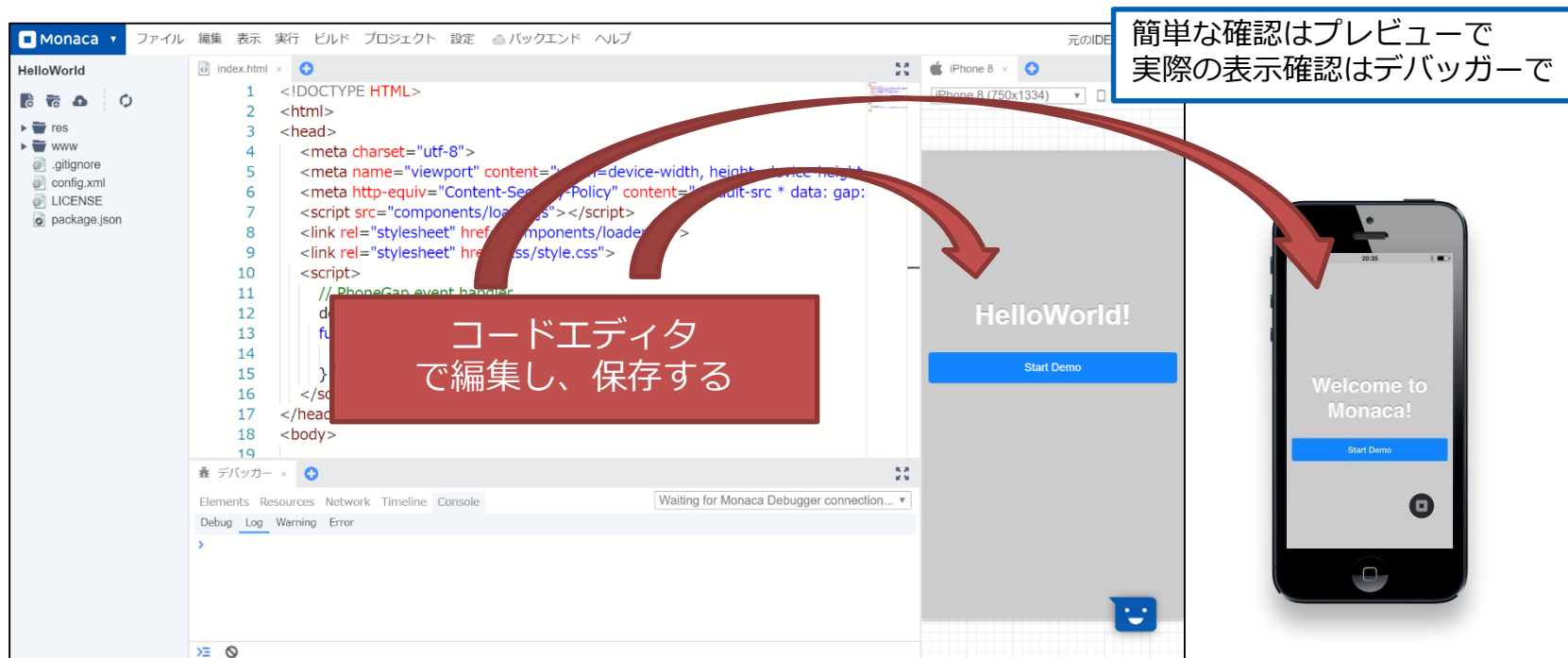
IDEの起動

IDEの各部の役割は以下の通りです。



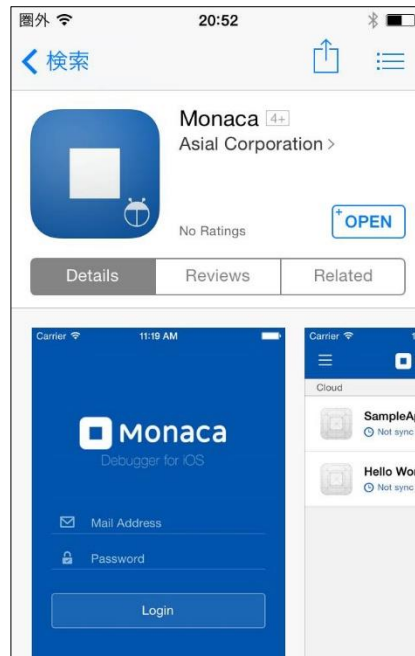
開発の流れ

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

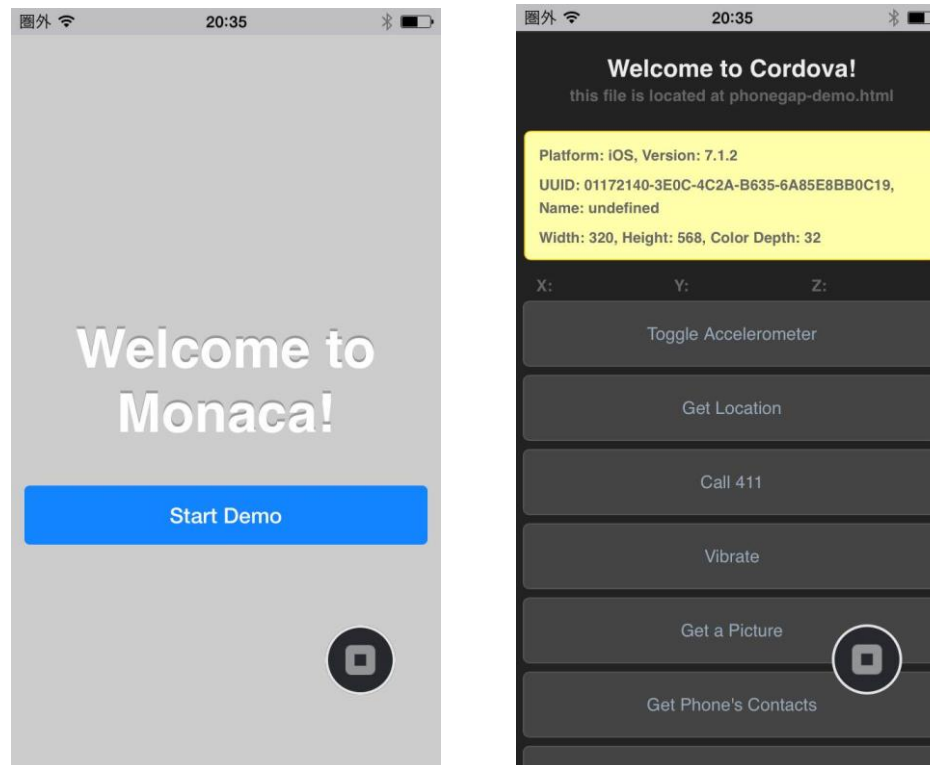
デバッガーを起動すると自動的にクラウドへ接続し、プロジェクトが一覧表示されます。

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。

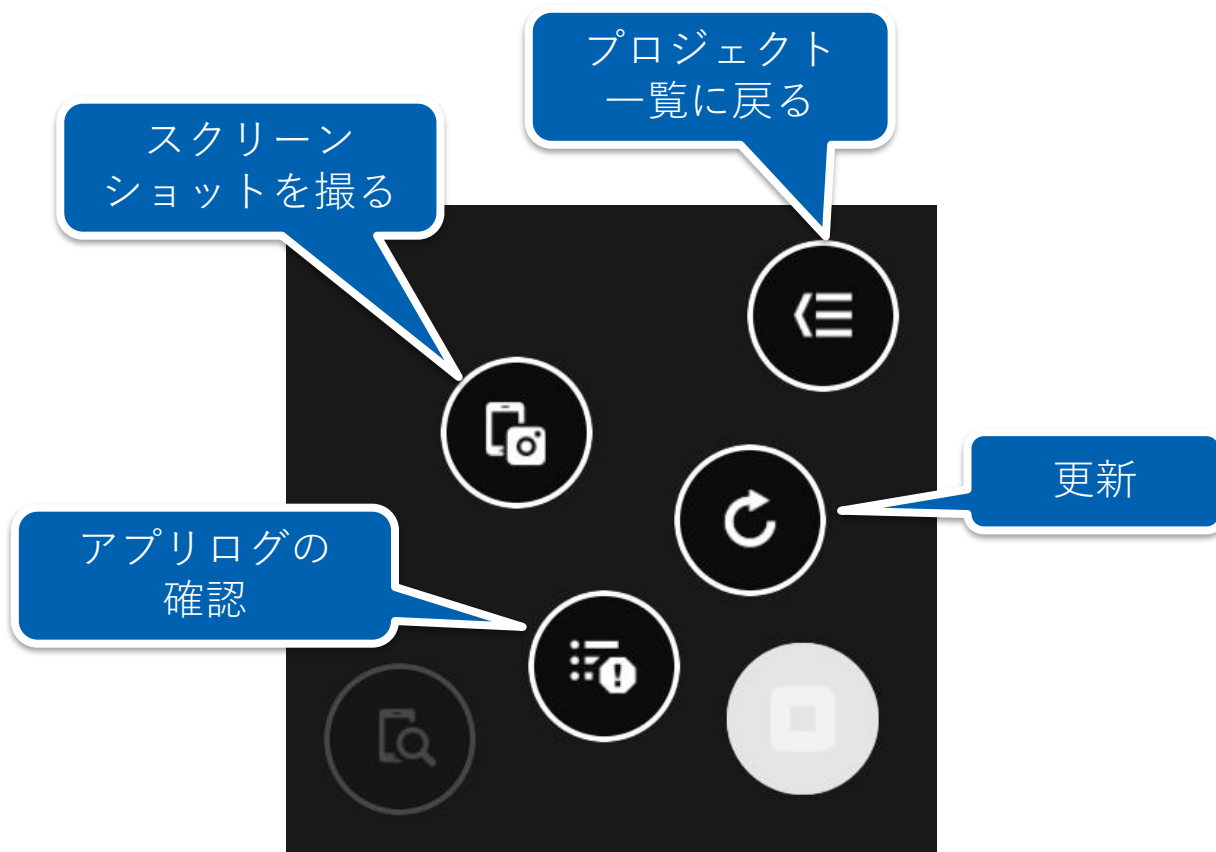


「Hello Worldアプリ」を動かしてみる

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

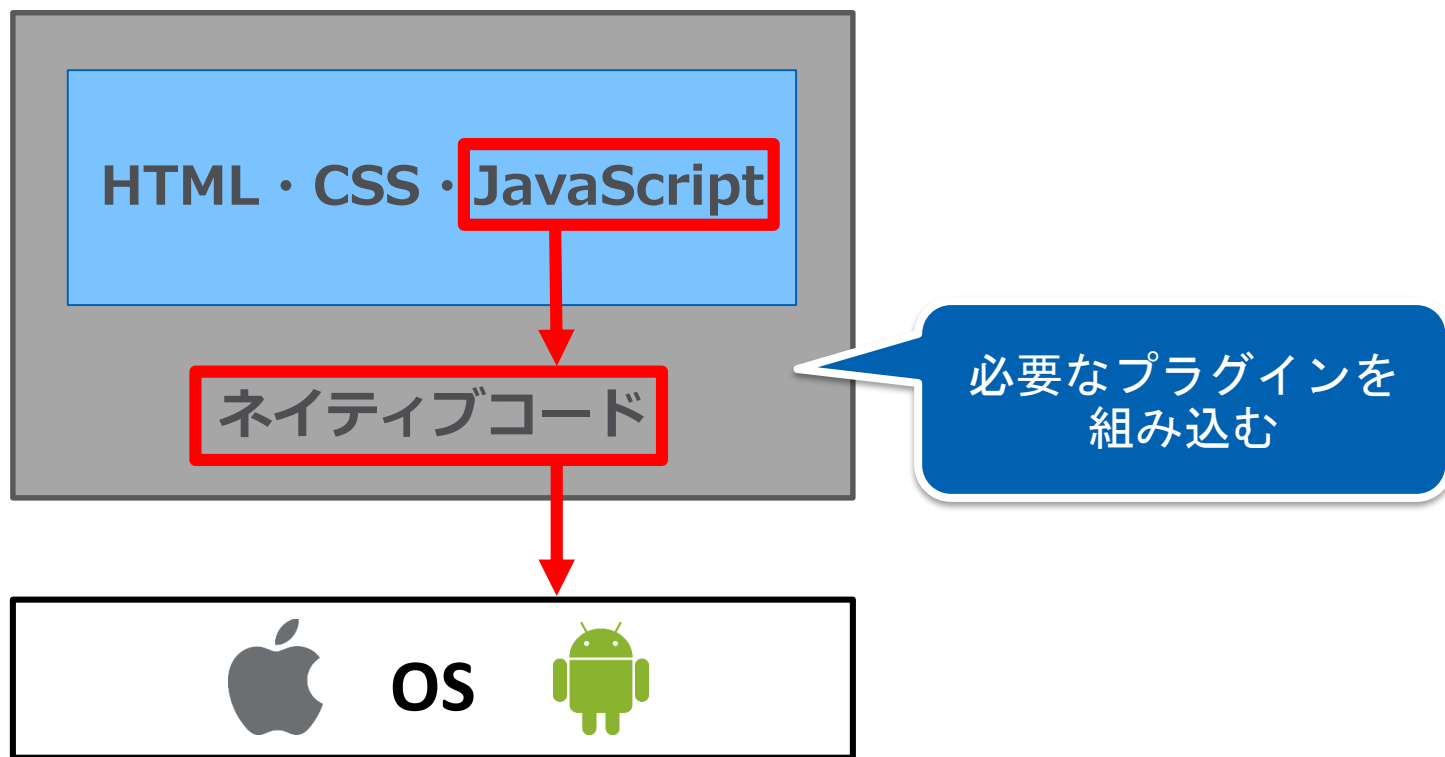


Cordovaプラグインについて



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

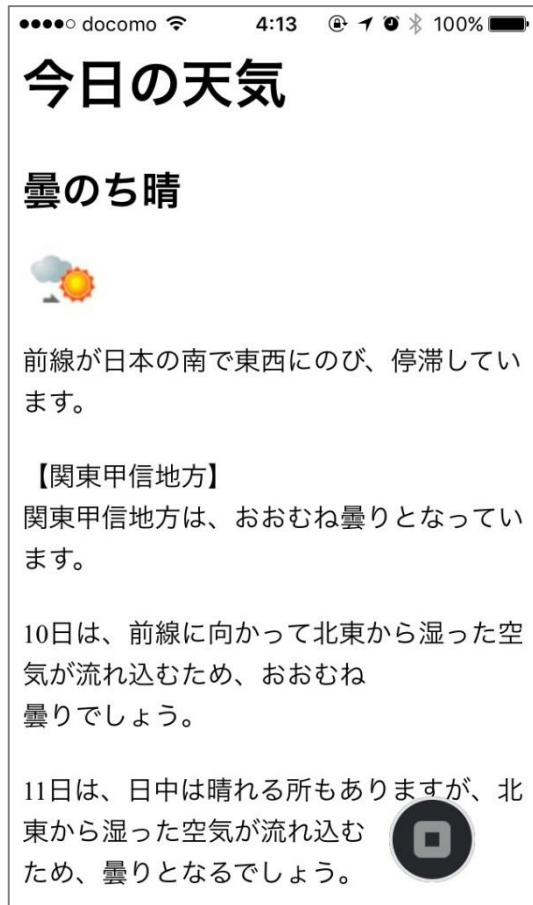
■ 第三者提供のプラグイン

- <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。

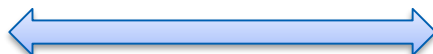
- Cordovaプラグイン
- JavaScript SDK
 - ・ クライアントサイド用に限る
 - ・ プロジェクトへのJSファイルのアップロード、またはCDNからの読み込みのどちらでも可
- Web API

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

The screenshot shows the Weather Hacks website. At the top, there's a header with the Livedoor logo and navigation links. Below the header, there's a main banner with the 'Weather Hacks' logo and a sub-header 'お天気Webサービス仕様'. The main content area is divided into two columns. The left column contains a section titled 'お天気Webサービス仕様' with a description of the service and a table of request parameters. The right column contains a 'メニュー' (Menu) section with links to various resources and a warning about RSS feed data.

livedoor 天気情報 livedoorが提供するお天気情報

ヘルプ livedoor

天気予報 災害情報

天気予報 スポット天気 Weather Hacks

トップ Weather Hacks > お天気Webサービス仕様

 **Weather Hacks**
ウェザーハックス

お天気Webサービス仕様

お天気Webサービス(Livedoor Weather Web Service / LWWS)は、現在全国142カ所の今日・明日・あさっての天気予報・予想気温と都道府県の天気概況情報を提供しています。

リクエストパラメータ

JSONデータをリクエストする際のベースとなるURLは以下になります。
<http://weather.livedoor.com/forecast/webservice/json/v1>
このURLに下の表のパラメータを加え、実際にリクエストします。

パラメータ名	説明
city	地域別に定義されたID番号を表します。 リクエストする地域とIDの対応は 全国の地点定義表(BSS) 内の「1次細分区分(cityタグ)」のidをご参照下さい。(例・佐賀県 伊万里=410020)

(例)「福岡県・久留米の天気」を取得する場合
下記URLにアクセスしてJSONデータを取得します。
基本URL + 久留米のID(400040)
<http://weather.livedoor.com/forecast/webservice/json/v1?city=400040>

メニュー

- ・ [お天気プラグイン](#)
- ・ [iCal天気](#)
- ・ [RSSフィード一覧](#)
- ・ [お天気Webサービス仕様](#)
- ・ [Weather Hacks用語集](#)
- ・ [Weather Hacks Q&A](#)

 livedoorが提供するRSSフィードデータは、予告なく変更・配信を終了する場合があります。あらかじめご了承下さい。

[免責事項](#)

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

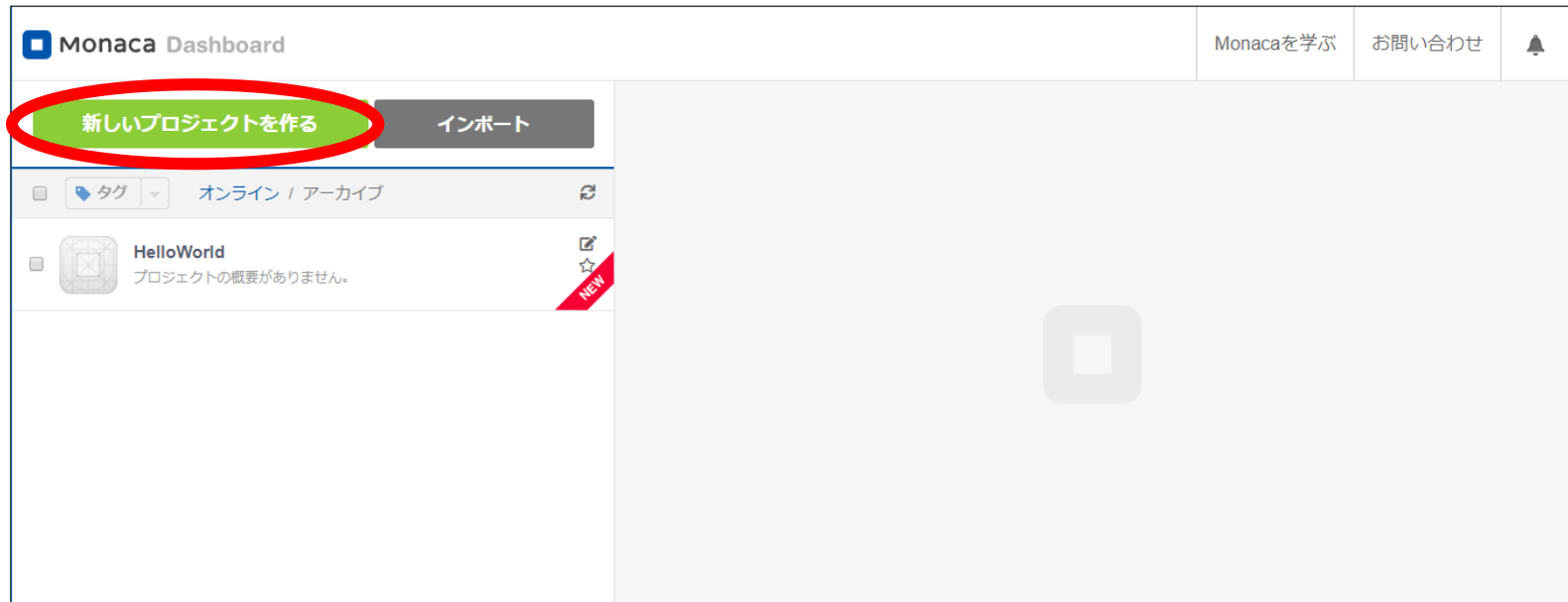
HTML5から追加された、標準的な非同期通信用のAPIです。現在では多くのブラウザでサポートされています。

アプリの作成



新しいプロジェクトを作成する

ダッシュボードに戻り、「新しいプロジェクトを作る」ボタンを選択します。



新しいプロジェクトを作成する

1. テンプレートの種類は「最小限のテンプレート」を選択します

☒ テンプレートの種類

サンプル アプリケーション サンプルのアプリケーションからプロジェクトを作成します。	フレームワーク テンプレート フレームワークが入っているテンプレートからプロジェクトを作成します。	最小限のテンプレート フレームワークを利用しない空白のプロジェクトを作成します。
---	--	---

2. プロジェクト名と説明（任意入力）を設定して作成します

2 プロジェクトの情報

プロジェクト名
カメラアプリ

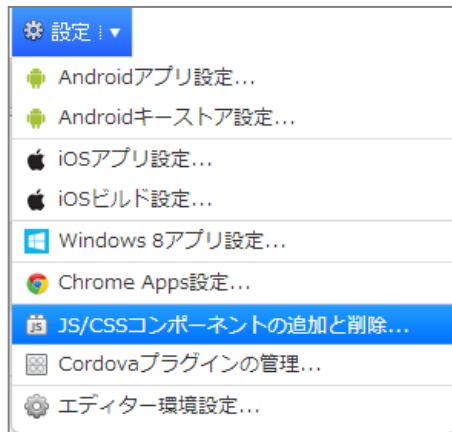
説明
お天気情報アプリ
に変更

作成

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。

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:

パッケージ名:

ビルド種別ごとに
パッケージ名を分ける: ☐ 有効

バージョン:

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: ☐ 有効

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。 Monaca Businessプラン以上が必要です。
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。

ビルド



Android デバッグビルド

Project v1.0.0

ビルド中...

ビルドログ

```
Adding android project...
Creating Cordova project for the Android platform:
  Path: platforms/android
  Package: com.example.helloworld
  Name: Minimum_Template
  Activity: MainActivity
  Android target: android-25
  Subproject Path: CordovaLib
  Android project created with cordova-android@6.2.3
```

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

【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して下さい。

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 ダッシュボード右上のユーザーアイコン > [プラン管理]を選択



- ② プラン管理画面で、「プラン変更」ボタンをクリック



プランのアップグレード

- ③ 利用プランの一覧から「アクティベーションコードを使う」を選択し、16桁のコードをハイフン付きで入力、「次に進む」ボタンを押すとプラン変更が完了します。

プラン変更

利用プラン選択 プランはいつでも変更できます。 [プランを比較する](#)

アクティベーションコードを使う

- コードをお持ちの方はこちらを選択してください。
- すぐにアップグレードが行われます。

アクティベーションコード

次に進む

XXXX-XXXX-XXXX-XXXX 形式で入力

サービス連携



外部サービスとの連携方法

■ Monacaプロダクトパートナー

<https://ja.monaca.io/product-partner/index.html>

バックエンドサービス、広告、分析、セキュリティ診断、テスト支援など、Monacaと連携可能な各種ツールが提供されています。

■ プロダクトパートナーに掲載がないサービスであっても、以下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
- クライアントサイド用のJavaScript SDK
- Web API

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/
- 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/
- 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/>