

プロフィールアプリを作ろう！



はじめに

今日の資料

<http://bit.ly/MonacaEasyStartApp>

教えてくれる人

池田 夏藻 (イケダ ナツモ)

富士通クラウドテクノロジーズ株式会社
ニフクラ mobile backend 企画チームに所属



ニフクラ mobile backend オリジナルキャラクター
タカノくん

何ができるの？

プロフィールアプリを作りましょう

出来上がったら作ったアプリを使って
みんなで自己紹介（じこしょうかい）
をしましょう👍

メッセージ機能をつけて
お友達のアプリにコメントしましょう💬



これからやること

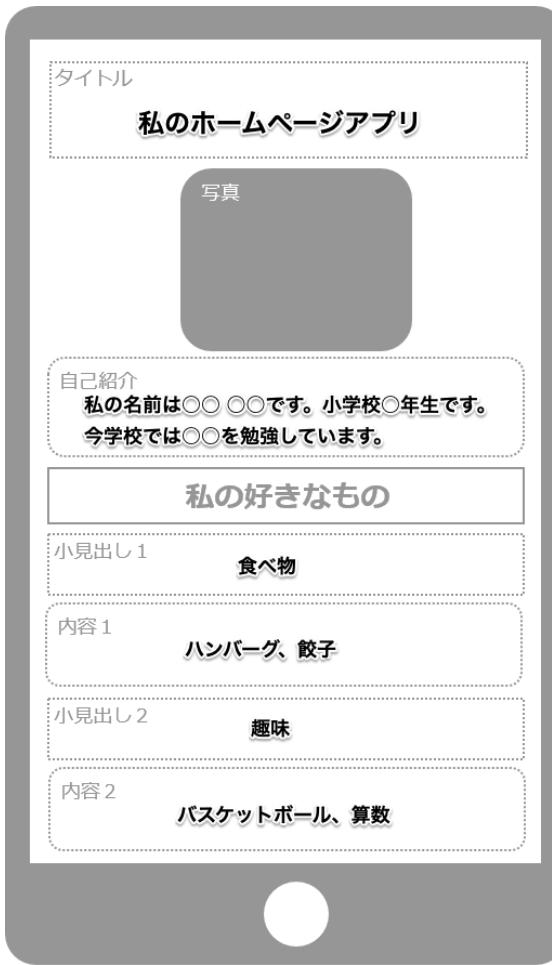
1. プロフィールアプリの下書き
2. アプリ作成ツール「Monaca」の準備
3. Monacaの使い方練習
4. プロフィールアプリを作る①（パーツの書き方）
5. プロフィールアプリを作る②（アレンジの仕方）
6. TRY：コメント入力機能をつけよう
 1. 機能追加に欠かせない道具 ニフクラ mobile backend とは
 2. ニフクラ mobile backend の準備
 3. コピペで機能追加
 4. 動作確認

1. プロフィールアプリの下書き

ワークシートにプロフィールアプリの下書きをしましょう。



こんな感じで書けていればOK！下書きの内容をアプリにしていきましょう。



2. アプリ作成ツール「Monaca」の準備

「**Monaca**」という開発ツールを使ってアプリを作ります。Monacaは子ども向けの開発ツールではありません！プロのエンジニアも使っているツールです。プログラミングには開発言語というコンピューター専用の言葉を使います。MonacaではHTML/JavaScript/CSSという比較的子どもでも取り組みやすい開発言語を使っています。今回は主に **HTML** の書き方を学びましょう😊



Monaca を使うために必要なもの

パソコンに用意されているもの

- Google Chrome (ブラウザ)
 - <https://www.google.com/chrome/>

持ってきてもらったもの

- Gmail (メールアドレス)
 - <https://accounts.google.com/signup>

アプリ作成ツール「Monaca」のアカウントを作る

Google Chrome をダブルクリックしてブラウザを開きます。



次のURLを入力してEnterキーを押します。

<https://monaca.mobi/ja/signup>



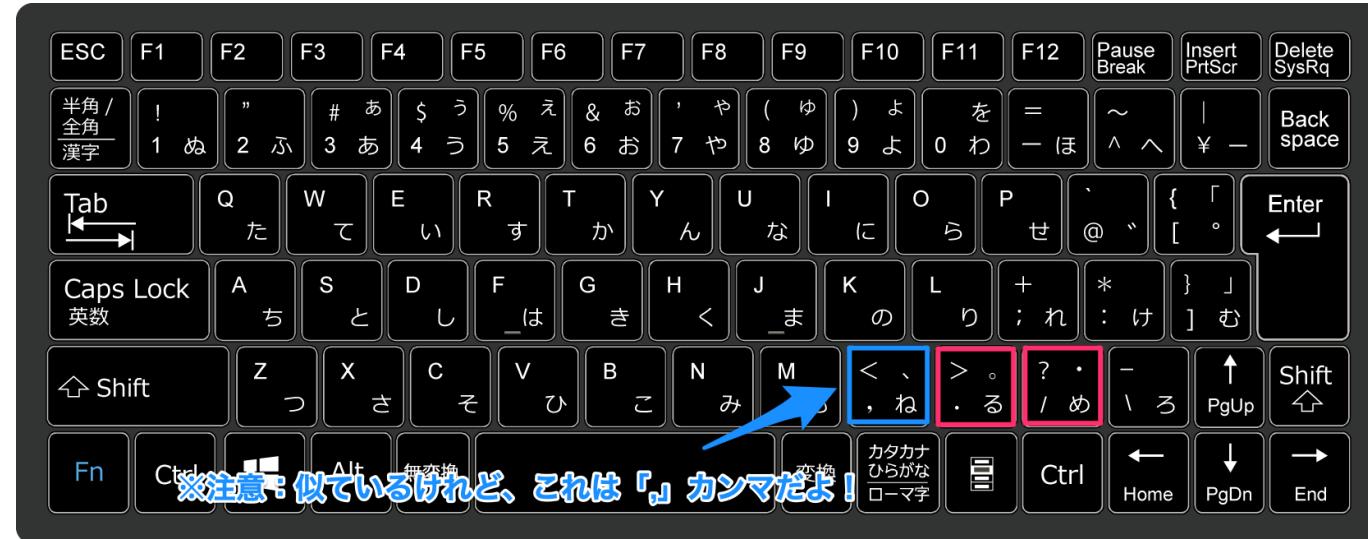
半角英語数字の入力をするには？

「ひらがな」が入力されてしまうときは、入力方法を切り替えるましょう。



記号はどこ？

URLに含まれる「.」や「/」はここにあるよ！



※注意: 似ているけれど、これは「.」カンマだよ!

アプリ作成ツール「Monaca」のアカウントを作る

Monacaのアカウント作成ページが開きます。

Gmailのメールアドレスとパスワード
(Gmailと同じでも違ってもどちらでも
OK) を入力して「アカウント新規作成」を
クリックします。

アカウント作成

アカウント作成

メールアドレス **【必須】** *Gmail を入力する*

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。
*パスワードを決めて
入力する*

アカウント新規作成

or

GitHubアカウントで作成

記号はどこ？

メールアドレスに含まれる「@」はここにあるよ！



アプリ作成ツール「Monaca」のアカウントを作る

Gmailにメールがとどきます。
とどいたメールを確認しましょう。



確認メールを送信しました

確認メールを [REDACTED] に送信しました。
送信された確認メール内に記載されたURLにアクセスし、登録の完了をお願いいたします。

再送する場合は、以下のボタンをクリックしてください。

[確認メールを再送する](#)

メールアドレスの入力を間違えた場合は、ログアウト後に正しいメールアドレスでご登録ください。

[ログアウト](#)

アプリ作成ツール「Monaca」のアカウントを作る

Google Chrome で別のタブを開きます。



次のURLを入力して Gmail を開きます。

<https://mail.google.com/>

Gmailを入力して、「次へ」をクリックします。

A screenshot of the Google Account sign-in page for Gmail. The page features the Google logo and the word 'ログイン' (Login). It asks users to use their 'お客様の Google アカウントを使用' (Use your Google account). A blue input field is labeled 'メールアドレスまたは電話番号' (Email address or phone number) with a red arrow pointing to it. Below the input field is the text 'Gmailを入力' (Enter Gmail). A blue link 'メールアドレスを忘れた場合' (Forgot email address) is also present. At the bottom, there is a note: 'ご自分のパソコンでない場合は、ゲストモードを使用して非公開でログインしてください。 詳細' (If you're not using your own computer, log in using Guest mode. Details). Two buttons are at the bottom: a blue 'アカウントを作成' (Create account) button and a blue '次へ' (Next) button.

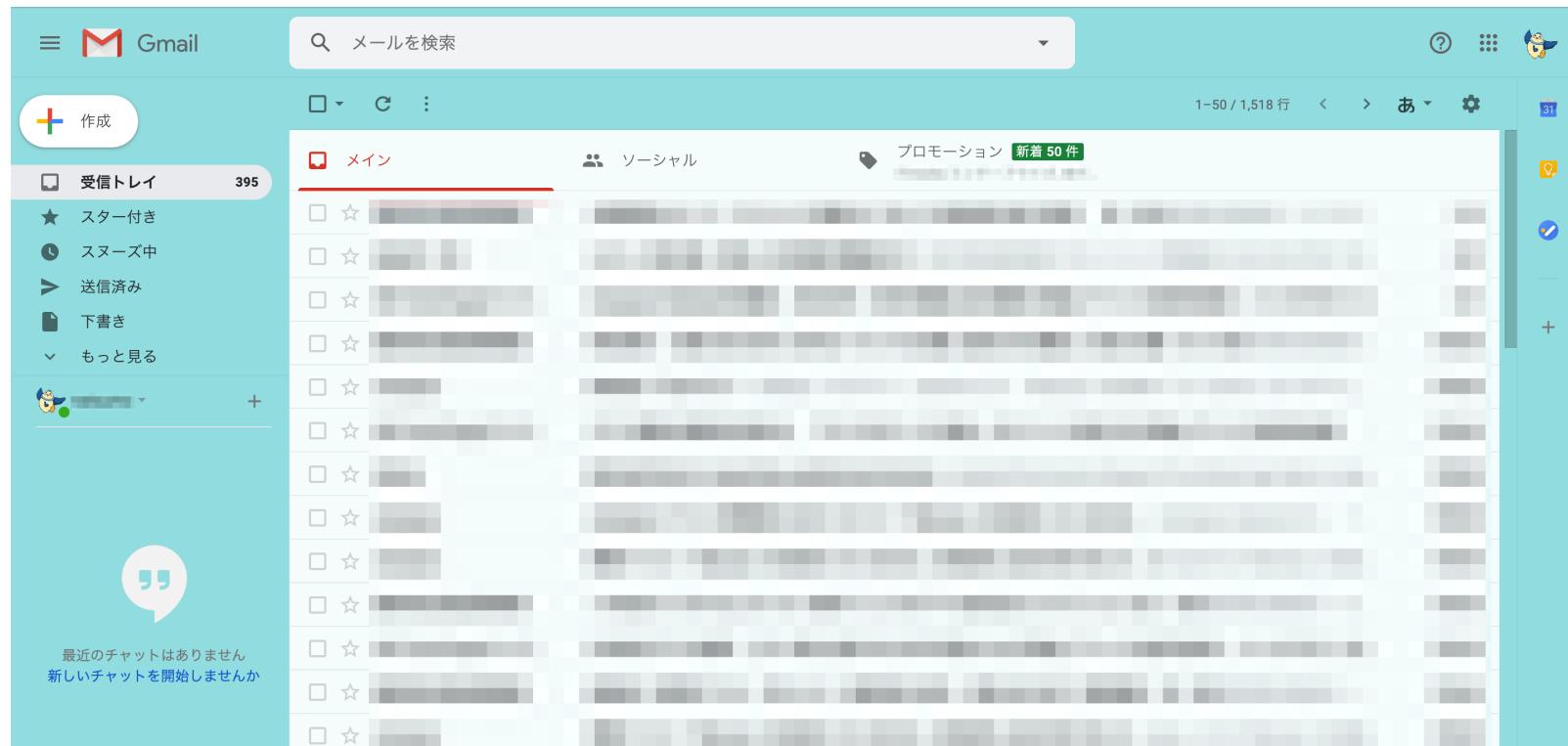
アプリ作成ツール「Monaca」のアカウントを作る

Gmailのパスワードを入力して、「次へ」をクリックします。



アプリ作成ツール「Monaca」のアカウントを作る

Gmailにログインできました。Monacaからのメールを確認しましょう。



アプリ作成ツール「Monaca」のアカウントを作る

Monacaからのメールを開いて「本登録はこちら」をクリックします。



Copyright © Asial Corporation. All rights reserved.

アプリ作成ツール「Monaca」のアカウントを作る

別のタブが開いて、Monacaの「無料トライアルを開始」という画面が表示されます。
「利用プラン選択」をクリックします。



無料トライアルを開始

14日の無償トライアルを開始します。トライアル中はいくつかの制限があります。
[アクティベーションコードをお持ちですか？](#)

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](#)

Businessプラン

- 法人・個人事業主で利用される方におすすめです。
- 導入サポート、技術サポートを提供いたします。



個人情報 お名前 【必須】

会社名 【必須】

電話番号

アプリ作成ツール「Monaca」のアカウントを作る

「Freeプラン」をクリックします。

Proプラン

- 個人で利用される方を対象にしています。
- 最新2世代のビルド環境のみ利用いただけます。
- 請求書支払いには対応していません。

Businessプラン

- 法人・個人事業主で利用される方におすすめです。
- 導入サポート、技術サポートを提供いたします。

Enterpriseプラン

- より安全・セキュアなアプリ開発に対応します。◆
- 請求書支払いをご利用いただけます。
- 最低5名分のライセンス購入が必要となります。

Freeプラン

- 個人で利用される方を対象にしています。
- 最新2世代のビルド環境のみ利用いただけます。
- ビルドに制限があります。(24時間以内に3回まで)

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。



アプリ作成ツール「Monaca」のアカウントを作る

「お名前」のみ入力して「無料トライアル開始」をクリックします。



無料トライアルを開始

14日の無償トライアルを開始します。トライアル中はいくつかの制限があります。
[アクティベーションコードをお持ちですか？](#)

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](#)

Freeプラン

- 個人で利用される方を対象にしています。
- 最新2世代のビルド環境のみ利用いただけます。
- ビルドに制限があります。(24時間以内に3回まで) **名前を入力する**

個人情報 お名前 **【必須】**

電話番号

無料トライアル開始

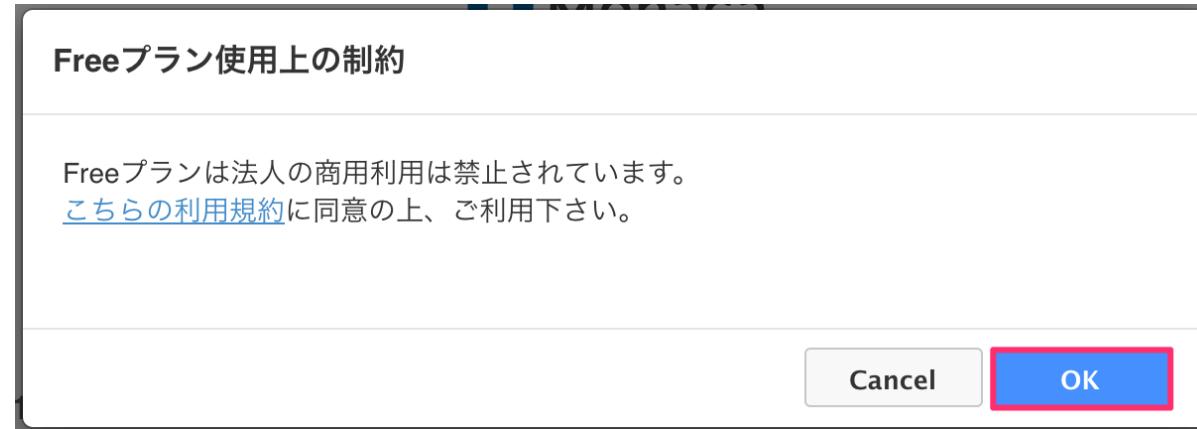
ひらがな・カタカナや漢字を入力するには？

半角英語数字入力からひらがな・カタカナや漢字の入力に変更するときは、もう一度キーを押しましょう。



アプリ作成ツール「Monaca」のアカウントを作る

「OK」をクリックします。



アプリ作成ツール「Monaca」のアカウントを作る

Monacaアカウントの作成は完了です

「ダッシュボードに進む」をクリックしま
しょう。



トライアルを開始しました！

さっそくMonacaではじめてのアプリを作ってみましょう。
ダッシュボードに進むと、ツアーが始まります。

プラン名

Freeプラン

トライアル期間

20■■年■■月■■日 ~ 20■■年■■月■■日 (14日間)

開発者

[REDACTED]

[ダッシュボードに進む](#)

アプリ作成ツール「Monaca」のアカウントを作る

ダッシュボードが表示されました。

The screenshot shows the Monaca Dashboard interface. At the top, there's a header with the Monaca logo, a search bar, and navigation links for 'Monacaを学ぶ' (Learn Monaca), 'お問い合わせ' (Contact), a notification bell icon, and a user profile icon. Below the header, there are two main sections: a 'Projects' section and a 'Posts from Monaca' section.

Projects Section: This section has a green button labeled '新しいプロジェクトを作る' (Create a new project) and a grey button labeled 'インポート' (Import). It displays a single project card for 'はじめてのMonacaアプリ' (My first Monaca app), which is marked as 'New'. The card includes a small icon of a smartphone, a star icon, and a link to 'No project description.'

Posts from Monaca Section: This section features a heading 'Posts from モナカアドレス' (Posts from Monaca Address). It lists three recent posts:

- Cordova Hook スクリプト入門** (Introduction to Cordova Hook Script)
2019-07-04
Cordova Hook Script is a feature provided by the Apache Cordova team. When using Cordova Plugins, you may need to implement some logic in your JavaScript code to interact with the plugin's native functionality. This post provides an introduction to how you can do that.
- Cordova 9.0 のサポートを開始** (Support for Cordova 9.0 begins)
2019-06-27
Monaca has started supporting Cordova 9.0. This post discusses the changes and what users can expect.
- 札幌初の共同開催！Monaca UG SAPPORO #3レポート** (Report on the first joint meeting in Sapporo! Monaca UG SAPPORO #3)
2019-06-24
A report on the third meeting of the Monaca User Group in Sapporo, which was held in collaboration with v-kansai. The report highlights the topics discussed and the atmosphere of the meeting.

At the bottom right of the dashboard, there's a blue button labeled 'チュートリアル' (Tutorial).

(参考) お家でMonacaを使うときは、<https://ja.monaca.io/> からログインして使いましょう ↗

プロジェクトを作る

「新しいプロジェクトを作る」をクリックします。



プロジェクトを作る

「最小限のテンプレート」をクリックします。

⊗ 新しいプロジェクトを作る



プロジェクトを作る

プロジェクト名に「プロフィールアプリ」と入力して「作成」をクリックします。



プロジェクトを作る

プロジェクトが作成されました。作成されたプロジェクトを開きましょう。作成されたプロジェクト「プロフィールアプリ」をクリックします。



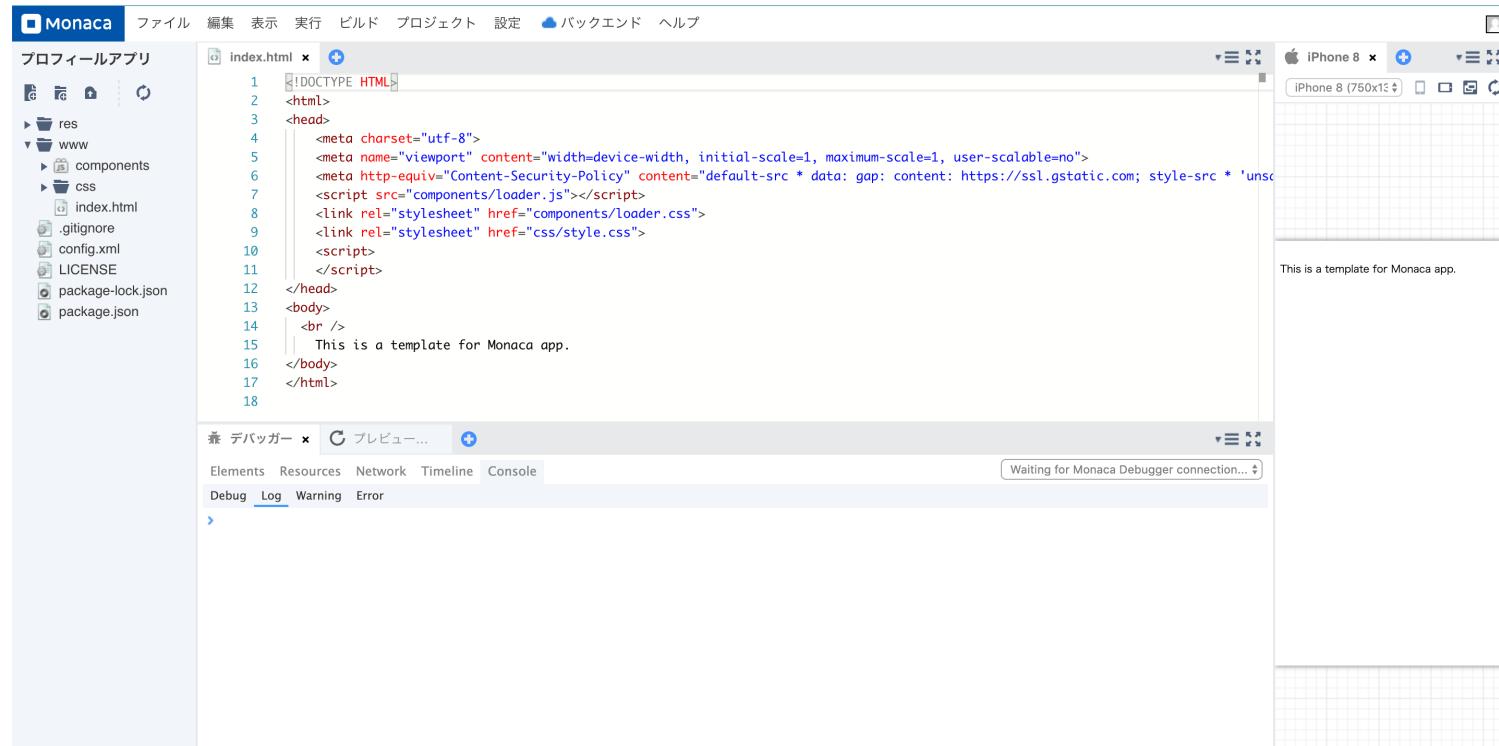
プロジェクトを作る

右に画面が開きます。「クラウドIDEで開く」をクリックします。

The screenshot shows the Monaca Dashboard interface. On the left, there's a list of projects: 'プロフィールアプリ' (Profile App) which is new, and two other projects whose names are blurred. On the right, a detailed view of the 'プロフィールアプリ' project is shown. It includes a summary section with 'プロジェクトの概要がありません' (No project summary), creation date '数秒前', last access '数秒前', and framework 'Cordova 9.0.0'. Below this are tabs for '開発' (Development), 'ビルド' (Build), 'デプロイ' (Deploy), and '設定' (Settings). Under the '開発' tab, there's a section titled 'クラウド開発' (Cloud Development) with a description: '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、ビルトといった開発に必要なすべての機能が備わっています。' (Monaca Cloud IDE is a development environment available exclusively via browser. It includes all the functions required for development such as coding, debugging, and building). A prominent blue button labeled 'クラウドIDEで開く' (Open in Cloud IDE) has a red arrow pointing to it. At the bottom, there's another section for 'コマンドラインツールを使う (ローカル開発)' (Use Command Line Tools (Local Development)).

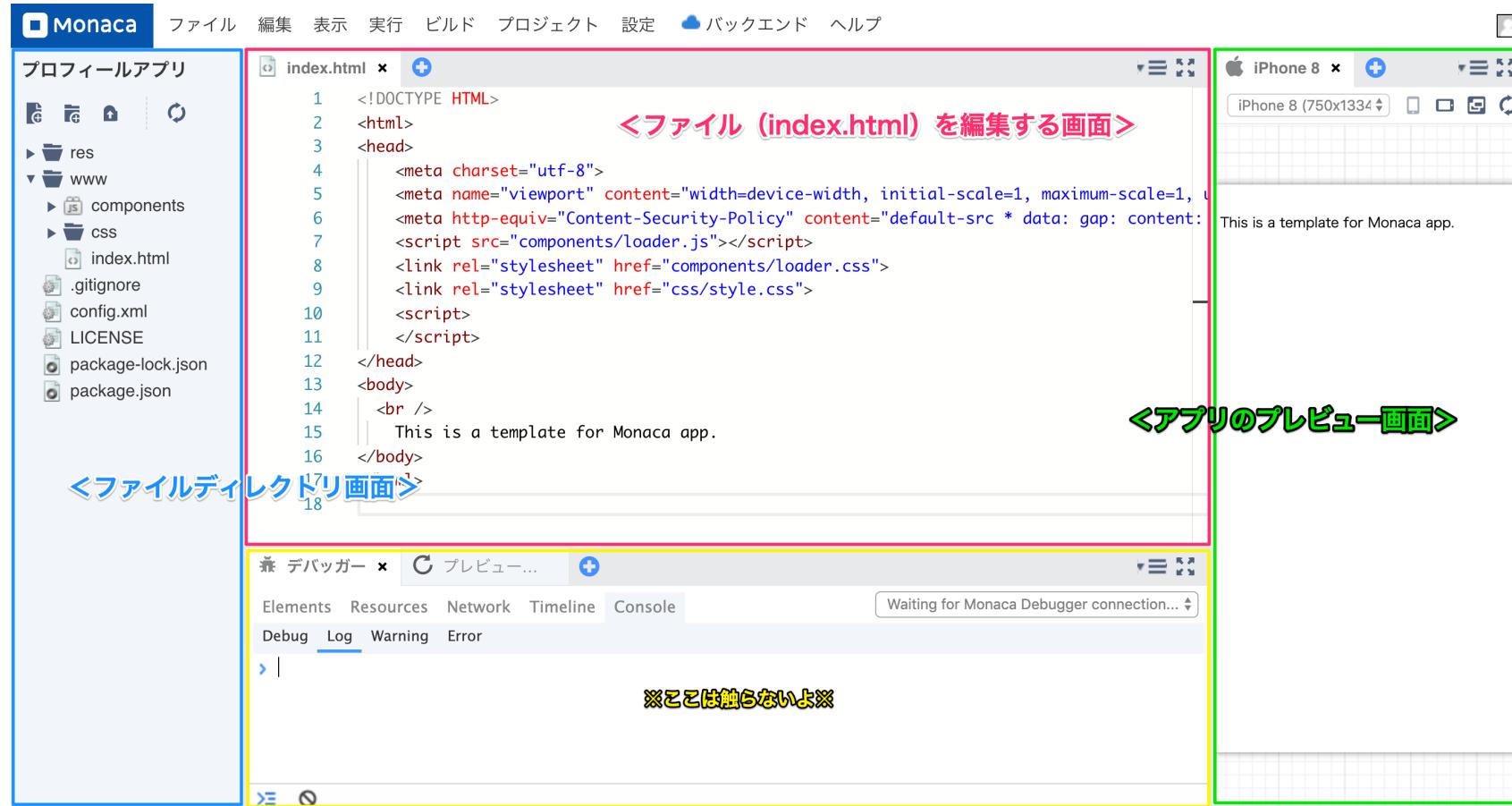
プロジェクトを作る

プロジェクトが開かれました。この画面を触って、アプリを作っていきます💪



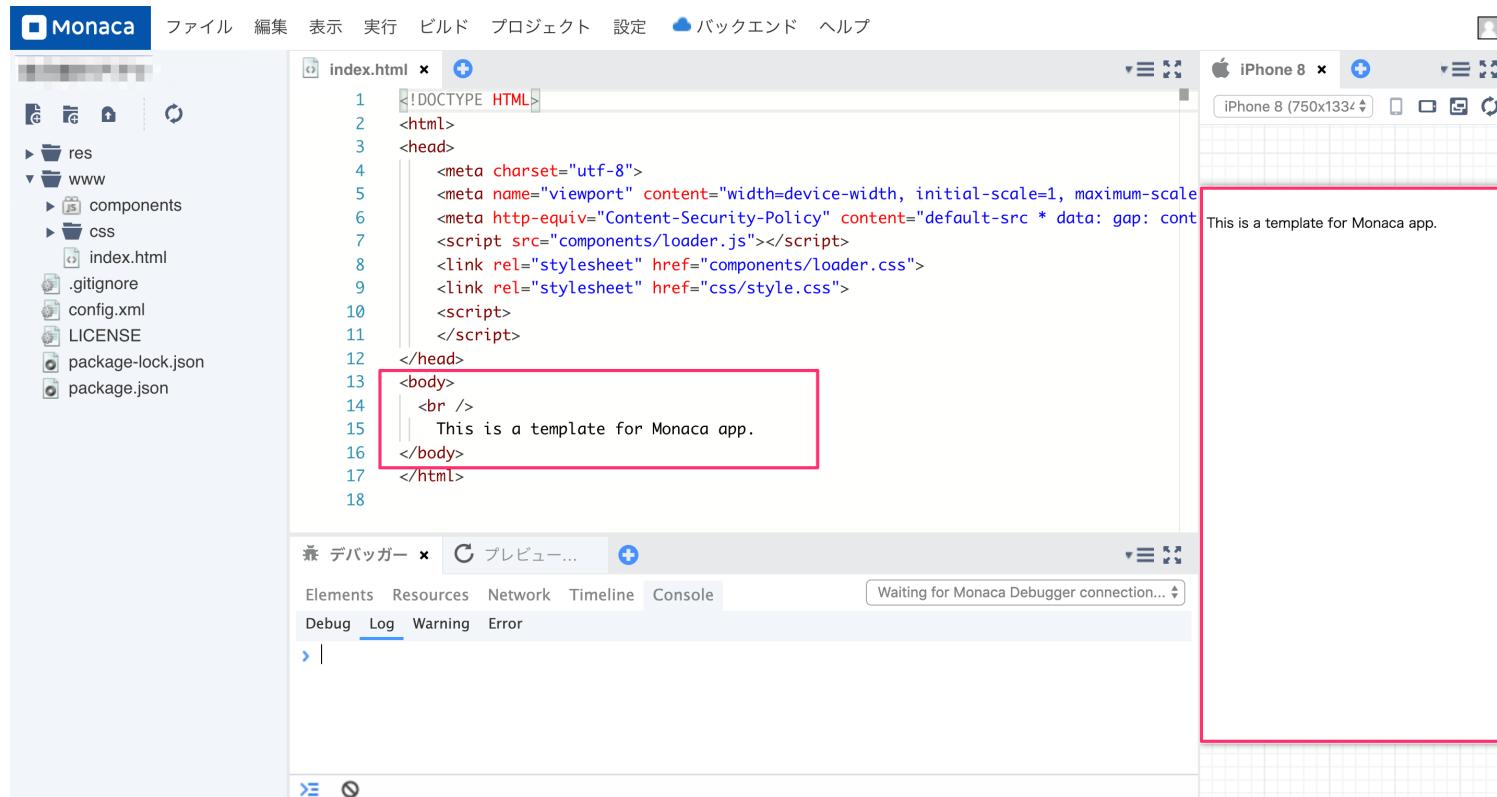
3. Monacaの使い方練習

Monacaの開発画面は次のようになっています。



使い方の練習をしていきましょう。編集画面に初めから開かれている **index.html** ファイルをみてください。

`<body>`と`</body>`の間に書かれた文字がプレビュー画面にも出ていますね。



<body>,</body> は「bodyタグ」と言います。

bodyタグの間に書いた内容がアプリの画面に表示される仕組みになっています。
一度bodyタグ内を全て消して書き直してみましょう。

```
9    ||  <link rel="stylesheet" href="css/style.css"
10   ||<script>
11   ||</script>
12  </head>
13  <body>
14  |<br>
15  </body>
16  </html>
17
```

Back Space キー
で消す

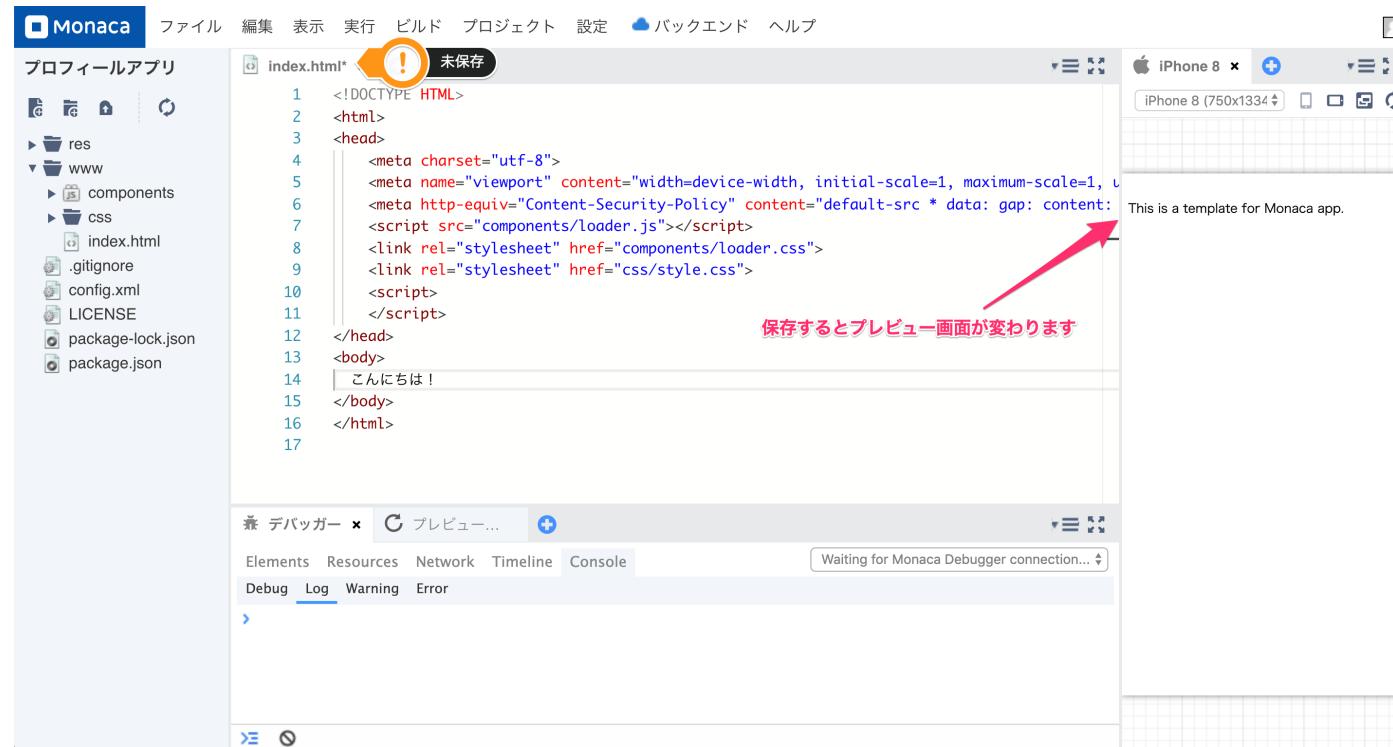


全部消したら「こんにちは！」と書いてみましょう。

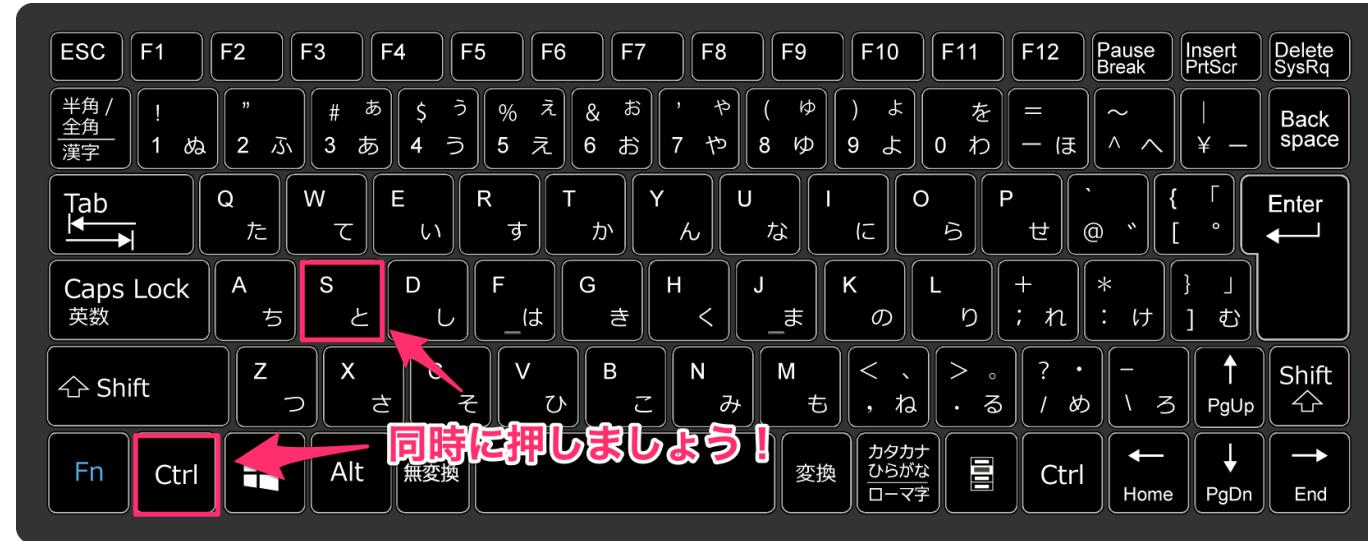
```
9      <link rel="stylesheet" href="css/style.css">
10     <script>
11     </script>
12   </head>
13   <body>
14   こんにちは！| 入力する
15 </body>
16 </html>
17
```

あれ？書き直したけどプレビュー画面が変わらないね😢

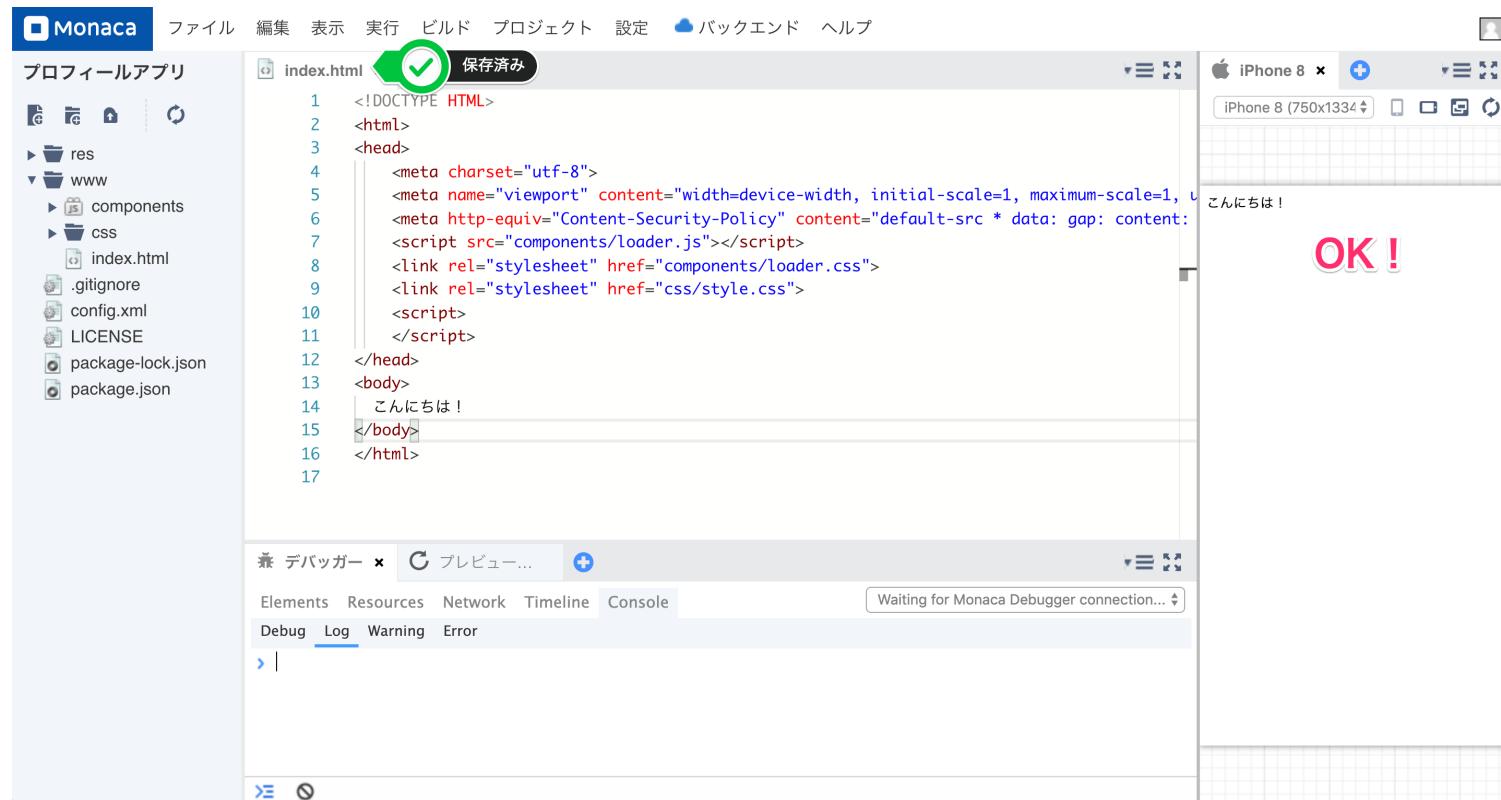
プレビュー画面に正しく表示させるには編集したファイル（index.html）を保存しなければなりません。



「Ctrl」 + 「S」 キーを同時に押して、保存をしてみましょう。



index.html が保存されて、プレビュー画面も更新されましたね 😊



練習はおしまいです！いよいよアプリを作っていきますよ 😊

4. プロフィールアプリを作る① (ページの書き方)

さっき下書きした内容をアプリに表示するように`<body>`と`</body>`の間に書き込んで行きましょう。

完成イメージはこんな感じです。



でもどうやって書いたらいいのかな？？書くために必要なパートを教えます ↗

見出し（タイトル・サブタイトル・小見出し）

- 見出しにしたい部分を h1タグ（`<h1>`, `</h1>`）で囲む
- h1タグで囲った見出しそれより小さい見出しえは h2タグ を使います
- h1, h2, h3, ...と見出しの大きさは 1 が一番大きく、数字が大きくなると小さくなります

```
<h1>見出し1</h1>
<h2>見出し2</h2>
<h3>見出し3</h3>
```

見出し（タイトル・サブタイトル・小見出し）

```
<h1>見出し1</h1>
<h2>見出し2</h2>
<h3>見出し3</h3>
```

段落（自己紹介文）

- 段落にしたい部分を pタグ（`<p>`, `</p>`）で囲む

```
<p>段落1</p>
<p>段落2</p>
<p>段落3</p>
```

段落（自己紹介文）

```
<p>段落1</p>
<p>段落2</p>
<p>段落3</p>
```

The screenshot shows the Xcode interface. On the left, the code editor displays the `index.html` file with the following content:

```
1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
6      <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: blob:>">
7      <script src="components/loader.js"></script>
8      <link rel="stylesheet" href="components/loader.css">
9      <link rel="stylesheet" href="css/style.css">
10     <script>
11     </script>
12 </head>
13 <body>
14     <p>段落1</p>
15     <p>段落2</p>
16     <p>段落3</p>
17 </body>
18 </html>
```

On the right, the iPhone 8 simulator window shows the rendered HTML with three paragraphs: "段落1", "段落2", and "段落3".

タグを使わないので書くとどうなるの？

改行（行がかわること）して書いても、
アプリ側では改行されずに表示されてしまいます 😞

The screenshot shows a code editor on the left and a mobile browser simulation on the right. The code editor displays the following HTML content:

```
index.html
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="Content-Security-Policy" content="default-src 'self' https://www.google-analytics.com/*>">
7   <script src="components/loader.js"></script>
8   <link rel="stylesheet" href="components/loader.css">
9   <link rel="stylesheet" href="css/style.css">
10  <script>
11  </script>
12 </head>
13 <body>
14  段落1
15  段落2
16  段落3
17 </body>
18 </html>
```

The mobile browser simulation on the right shows the rendered content with three paragraphs: "段落1", "段落2", and "段落3".

見出しと段落を組み合わせて、下書きした内容を画面の中に作ってみましょう！

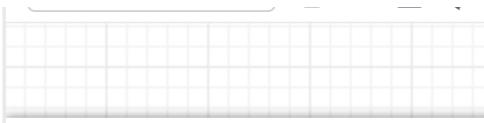


見出しと段落を組み合わせて、下書きした内容を画面の中に作ってみましょう！

```
<h1>私のホームページアプリ</h1>
<p>私の名前は〇〇　〇〇です。小学校〇年生です。今学校では〇〇を勉強しています。</p>
<h2>私の好きなもの</h2>
<h3>食べ物</h3>
<p>ハンバーグ・餃子</p>
<h3>趣味</h3>
<p>バスケ、算数</p>
```

見出しと段落を組み合わせて、下書きした内容を画面の中に作ってみましょう！

```
/   <script src="components/loader.js"></script>
8   <link rel="stylesheet" href="components/loader.css">
9   <link rel="stylesheet" href="css/style.css">
10  <script>
11  </script>
12 </head>
13 <body>
14 <h1>私のホームページアプリ</h1>
15 <p>私の名前は〇〇 〇〇です。小学校〇年生です。今学校では〇〇を勉強しています。</p>
16 <h2>私の好きなもの</h2>
17 <h3>食べ物</h3>
18 <p>ハンバーグ、餃子</p>
19 <h3>趣味</h3>
20 <p>バスケ、算数</p>
21 </body>
22 </html>
23
```



私のホームページアプリ

私の名前は〇〇 〇〇です。小学校〇年生です。今学校では〇〇を勉強しています。

私の好きなもの

食べ物

ハンバーグ、餃子

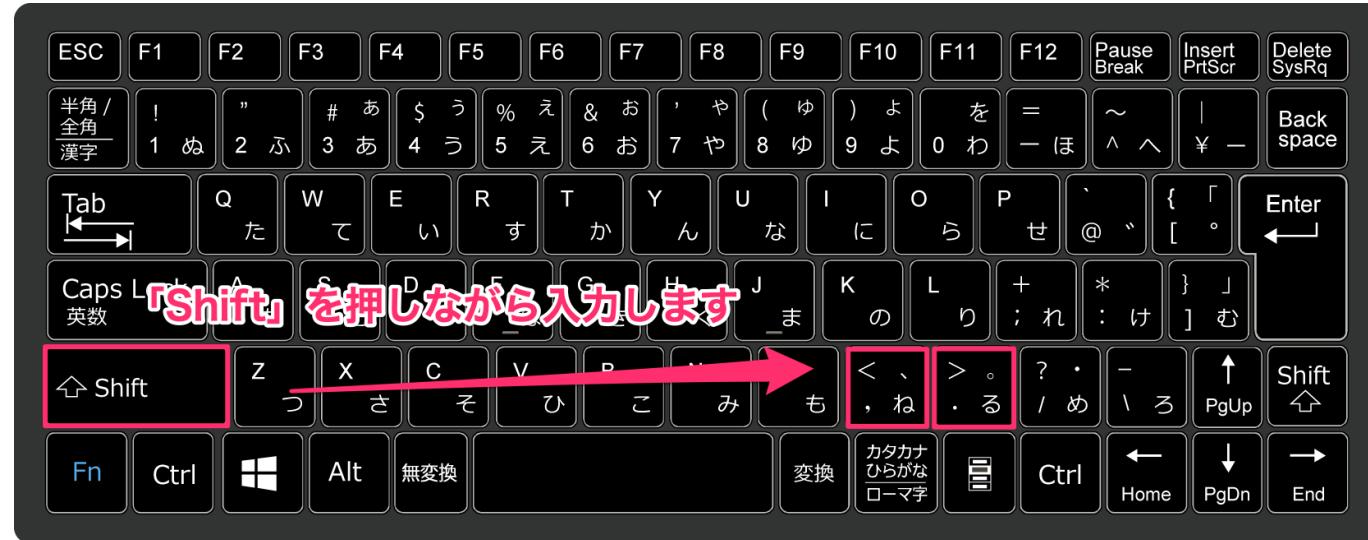
趣味

バスケ、算数

記号はどこ？

タグ含まれる「<」，「>」はここにあるよ！

ボタンの上の行に書かれている記号を入力するときは「Shift」ボタンを押しながら入力します。



画像

文字の入力が終わったら、画像を入れてみましょう！（ちょっと難しくなるよ👉👉）



画像

画像の準備

ブラウザで別のタブを開いて、好きな画像を探しましょう。
画像が決まったら画面に表示しておきましょう。

<例>

- http://bit.ly/mb_takano をブラウザの別のタブで開く
- タカノくんの画像が表示される



画像

画像の準備

画像の上で右クリックをして、「名前を付けて画像を保存」をクリックします。



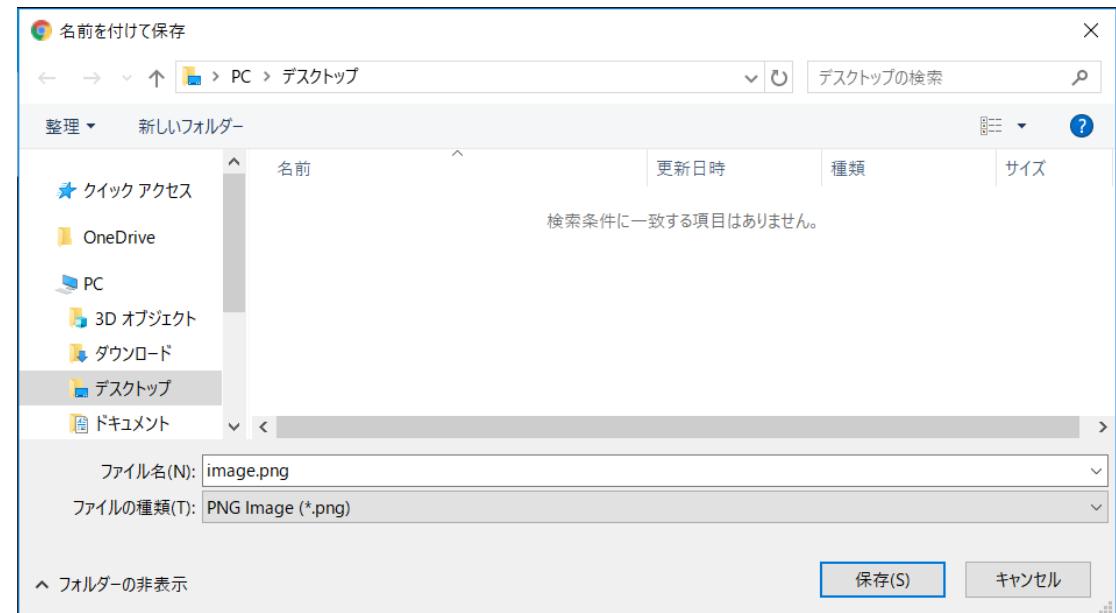
画像

画像の準備

保存画面が開いたら、保存場所とファイル名を次のようにしましょう。

- 保存場所 「デスクトップ」
- ファイル名 「image.png」

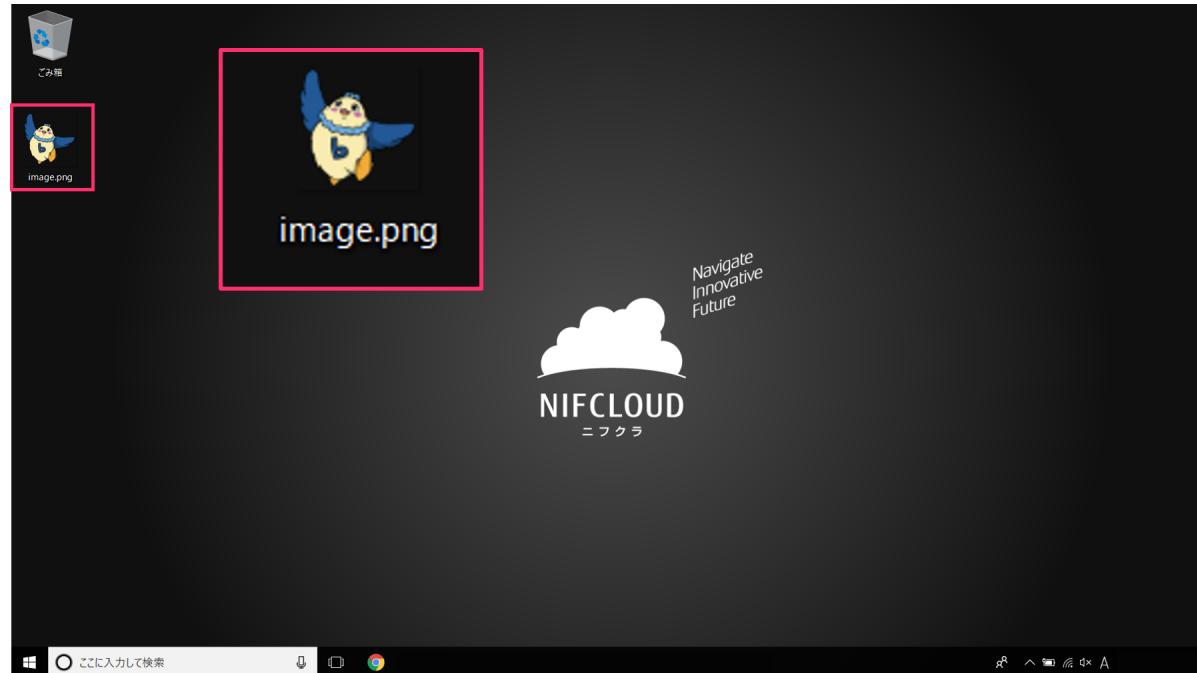
できたら 「保存」 をクリックします。



画像

画像の準備

デスクトップを確認しましょう。画像が表示されていればOKです。

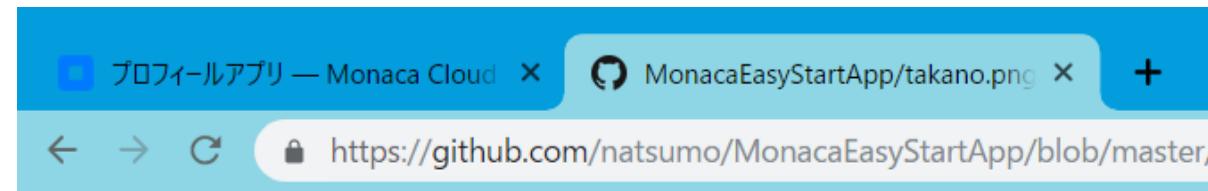


画像

Monacaへのインポート

用意した画像をアプリで表示するために、Monacaにインポート（Monacaに取り込むこと）しましょう。

ブラウザで先ほど開いていた**Monacaの開発画面のタブ**を開きます。
(先ほどの画面に戻ります。)

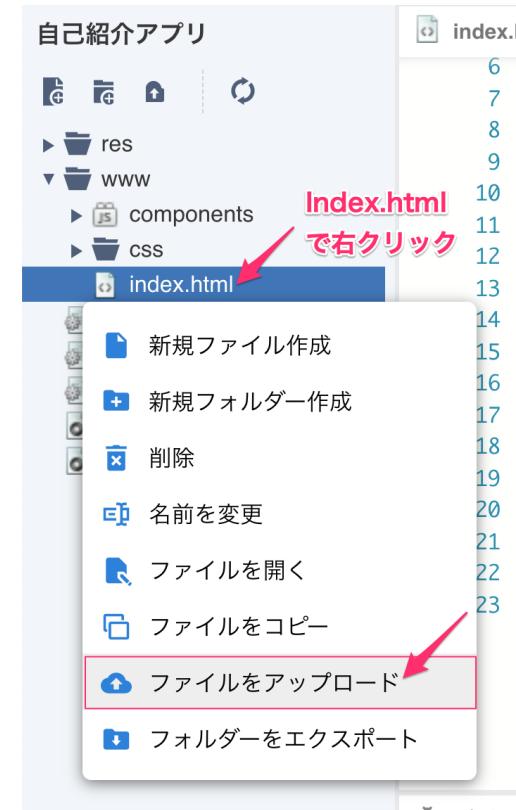


画像

Monacaへのインポート

編集している index.html と同じ場所
に画像をインポートしましょう。

ファイルディレクトリ画面の
index.html で**右クリック**します。
開いた項目の「ファイルをアップロード」
をクリックします。



画像

Monacaへのインポート

デスクトップに用意した画像をファイルアップロード画面に**ドラッグ&ドロップ**してファイルをアップロードします。

(参考) マウスでクリックした状態のままファイルを移動して、指定の場所でクリックを解除（指を離す）することを**ドラッグ&ドロップ**と言います。



画像

Monacaへのインポート

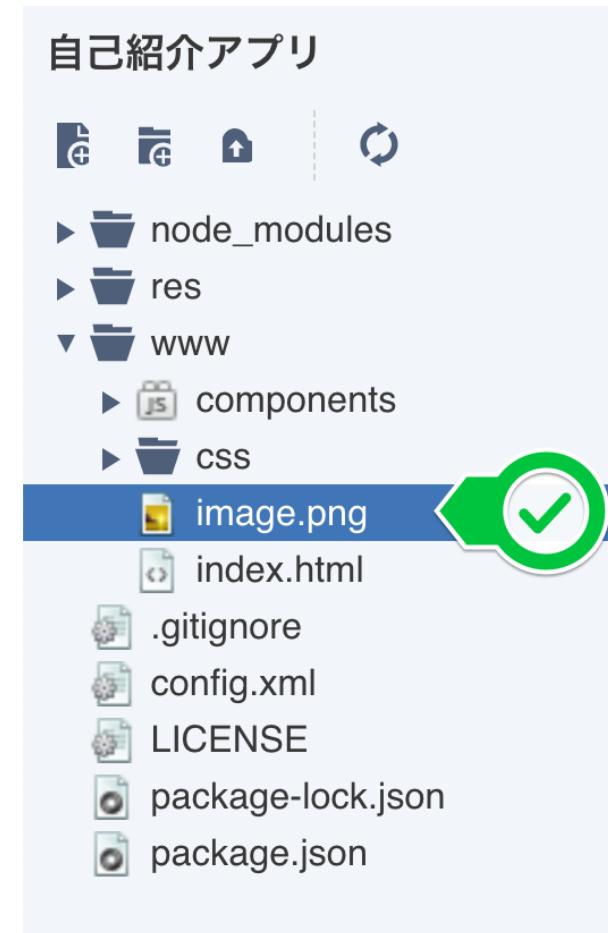
正しくファイルがアップロードされるとディレクトリに追加されます。追加されていることを確認したらアップロード画面を「x」をクリックして閉じましょう。



画像

Monacaへのインポート

改めてファイルディレクトリ画面を確認しましょう。
インポート出来たことを確認できました。



画像

インポートした画像をアプリに表示するには `img` タグを使います。

- `src` : 画像のパスを指定します
- `width` : 画像の横の長さを指定します
- `height` : 画像の縦の長さを指定します

```

```

画像

```

```



```
index.html x + iPhone 8 x + iPhone 8 (750x1334) + iPhone 8

2  <html>
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
6      <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: blob: " >
7      <script src="components/loader.js"></script>
8      <link rel="stylesheet" href="components/loader.css">
9      <link rel="stylesheet" href="css/style.css">
10     <script>
11     </script>
12 </head>
13 <body>
14     
15 </body>
16 </html>
17
```

プロフィールアプリに画像を表示してみましょう。タイトル（h1タグ）の下に追加してみましょう。

```
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1</meta>
6 <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content:>
7 <script src="components/loader.js"></script>
8 <link rel="stylesheet" href="components/loader.css">
9 <link rel="stylesheet" href="css/style.css">
10 <script>
11 </script>
12 </head>
13 <body>
14 <h1>私のホームページアプリ</h1>
15 
16 <p>私の名前は○○ ○○です。小学校○年生です。今学校では○○を勉強しています。</p>
17 <h2>私の好きなもの</h2>
18 <h3>食べ物</h3>
19 <p>ハンバーグ、餃子</p>
20 <h3>趣味</h3>
21 <p>バスケ、算数</p>
22 </body>
23 </html>
24
```



私のホームページアプリ



私の名前は○○ ○○です。小学校○年生です。今学校では○○を勉強しています。

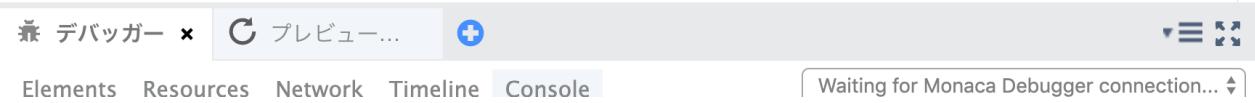
私の好きなもの

食べ物

ハンバーグ、餃子

趣味

バスケ、算数



記号はどこ？

imgタグに含まれる「"」, 「=」はここにあるよ！



「Shift」を押しながら入力します

5. プロフィールアプリを作る②^(アレンジの仕方)

ここまでで下書き通りには表示できるようになりました！

ですが、このままだとちょっと殺風景ですね…
せっかくなのでもう少し**アレンジ**してみましょう！

私のホームページアプリ



私の名前は〇〇 〇〇です。小学校〇年生です。今学校では〇〇を勉強しています。

私の好きなもの

食べ物

ハンバーグ、餃子

趣味

バスケ、算数

背景の色変更

- bodyタグに `bgcolor="色"` を追加します
- 下の例は背景を水色（skyblue）にします

```
<body bgcolor="skyblue">  
  ...  
</body>
```

背景の色変更

```
<body bgcolor="skyblue">  
    ...  
</body>
```

```
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
6 <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: cont
7 <script src="components/loader.js"></script>
8 <link rel="stylesheet" href="components/loader.css">
9 <link rel="stylesheet" href="css/style.css">
10 <script>
11 </script>
12 </head>
13 <body bgcolor="skyblue"> ↓
14 <h1>私のホームページアプリ</h1>
15 
16 <p>私の名前は○○○○です。小学校○年生です。今学校では○○を勉強しています。</p>
17 <h2>私の好きなもの</h2>
18 <h3>食べ物</h3>
19 <p>ハンバーグ、餃子</p>
20 <h3>趣味</h3>
21 <p>バスケ、算数</p>
22 </body>
23 </html>
```



背景の色変更

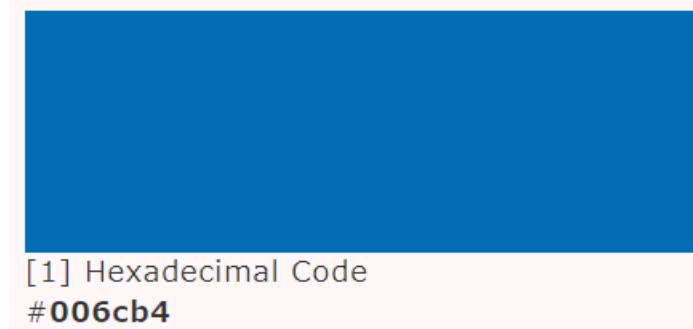
```
<body bgcolor="skyblue">  
...  
</body>
```

色は、青：blue, ピンク：pink, 赤：red など、
色名の英語表記または色コードで指定することができるよ ↗

＜例＞ タカノ君の羽の青色： #006cb4

参考リンク

WEB色見本 原色大辞典 - HTMLカラーコード
<https://www.colordic.org/>



文字の色変更

- 色を変えたい文字を fontタグ（``, ``）で囲む
- 下の例は赤文字の「こんにちは」を表示します

```
<font color="red">こんにちは</font>
```

文字の色変更

```
<font color="red">こんにちは</font>
```

```
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
6   <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: blob: none">
7   <script src="components/loader.js"></script>
8   <link rel="stylesheet" href="components/loader.css">
9   <link rel="stylesheet" href="css/style.css">
10  <script>
11  </script>
12 </head>
13 <body>
14 | <font color="red">こんにちは </font>
15 </body>
16 </html>
17
```

色の指定方法は背景色と同じく英語表記か色コードで指定できるよ ↗

文字の色変更

<例> タイトルの一部だけを変えることもできるよ👍

```
<h1>私の<font color="red">ホームページ</font>アプリ</h1>
```

```
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=1">
6   <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content-script-src 'unsafe-eval' components/loader.js components/loader.css 'unsafe-inline' style.css">
7   <script src="components/loader.js"></script>
8   <link rel="stylesheet" href="components/loader.css">
9   <link rel="stylesheet" href="css/style.css">
10  <script>
11  </script>
12 </head>
13 <body bgcolor="skyblue">
14 <h1>私の<font color="red">ホームページ</font>アプリ</h1>
15 
16 <p>私の名前は○○ ○○です。小学校○年生です。今学校では○○を勉強しています。</p>
17 <h2>私の好きなもの</h2>
18 <h3>食べ物</h3>
19 <p>ハンバーグ、餃子</p>
20 <h3>趣味</h3>
21 <p>バスケ、算数</p>
22 </body>
23 </html>
24
```

「ホームページ」だけ赤文字に



位置の変更

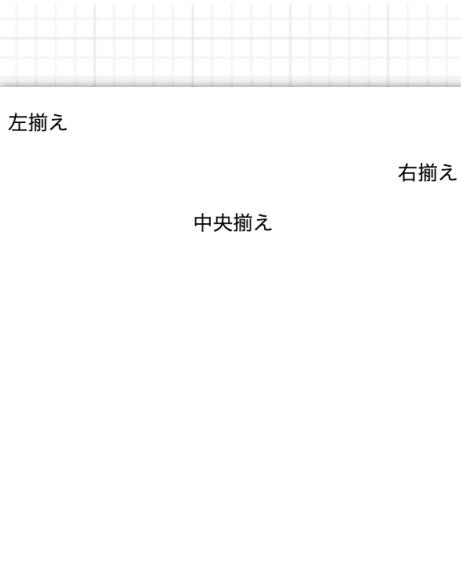
- 位置を変更したいタグにalignを追加します
- 下の例は段落（pタグ）の位置を変更します

```
<p align="left">左揃え</p>
<p align="right">右揃え</p>
<p align="center">中央揃え</p>
```

位置の変更

```
<p align="left">左揃え</p>
<p align="right">右揃え</p>
<p align="center">中央揃え</p>
```

```
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
6 <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: none">
7 <script src="components/loader.js"></script>
8 <link rel="stylesheet" href="components/loader.css">
9 <link rel="stylesheet" href="css/style.css">
10 <script>
11 </script>
12 </head>
13 <body>
14 <p align="left">左揃え</p>
15 <p align="right">右揃え</p>
16 <p align="center">中央揃え</p>
17 </body>
18 </html>
19
```



位置の変更

画像の位置変更をしたい場合

- 画像 (imgタグ) を pタグ (`<p align="***">, </p>`) で囲みましょう。

```
<p align="center">画像</p>
```

位置の変更

画像の位置変更をしたい場合

<p align="center">画像</p>

```
 5 <meta charset="utf-8">
 6 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
 7 <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content:">
 8 <script src="components/loader.js"></script>
 9 <link rel="stylesheet" href="components/loader.css">
10 <link rel="stylesheet" href="css/style.css">
11 <script>
12 </script>
13 </head>
14 <body bgcolor="skyblue">
15   <h1>私の<font color="red">ホームページ</font>アプリ</h1>
16   <p align="center"></p>
17   <p>私の名前は○○ ○○です。小学校○年生です。今学校では○○を勉強しています。</p>
18   <h2>私の好きなもの</h2>
19   <h3>食べ物</h3>
20   <p>ハンバーグ、餃子</p>
21   <h3>趣味</h3>
22   <p>バスケ、算数</p>
23 </body>
24 </html>
```



いくつか画像を追加したり、色を変えたりしてプロフィールアプリをアレンジしてみましょう💪



6. TRY : コメント入力機能をつけよう

最後に特別な機能を追加しましょう😊！

できたプロフィールアプリをみんなに見せてコメントをもらえるようにしましょう😍

この機能の追加はちょっと難しいのでコピペ（コピー＆ペースト（貼り付ける）のこと）を使ってサクッと機能を追加しちゃいましょう。

これからやること

1. 機能追加に欠かせない道具 **ニフクラ**

mobile backend とは

2. ニフクラ mobile backend の準備

3. コピペで機能追加

4. 動作確認



6.1. 機能追加に欠かせない道具ニフクラ mobile backend とは

スマートフォンアプリのクラウドデータベースサービスです。

と言っても難しくてよくわからないですよね... 😱

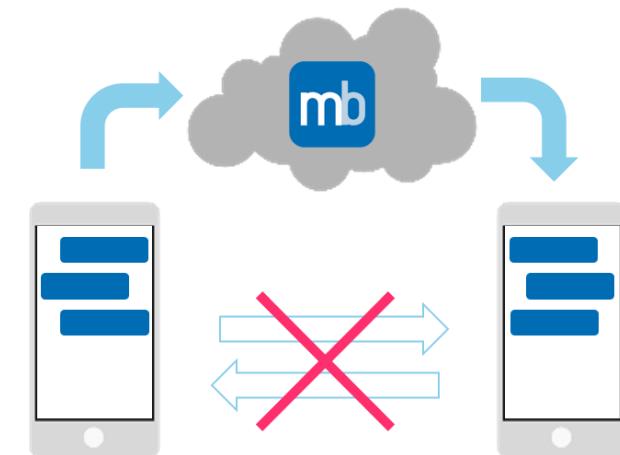
よくわからなくとも実はアプリに欠かすことのできない存在なんです。

<例>メッセージ送信アプリ

送信したメッセージはスマホからスマホへ送られ
ているわけではありません！

一度クラウドに保存され相手に届けられる仕組み
になっています。

クラウドがなければ作ることができないわけです
ね。



6.1. 機能追加に欠かせない道具ニフクラ mobile backend とは



✓ メッセージは管理画面
から確認できます

これから作る「コメント入力機能」もクラウドを使って作ってみましょう。
一人一人それぞれのクラウドを用意しますので、作ったアプリと一緒にメッセージデータが入ったクラウドも持ち帰えってもらえます👍

6.2. ニフクラ mobile backend の準備

ニフクラ mobile backend の利用登録をしましょう。

ブラウザで別のタブを開いて、次のURLを入力してEnterキーを押します。

<https://mbaas.nifcloud.com/>



6.2. ニフクラ mobile backend の準備

「無料登録」をクリックします。



6.2. ニフクラ mobile backend の準備

SNS ID として Gmail を使ってアカウントを作成します。
「会員登録（無料）」をクリックします。



6.2. ニフクラ mobile backend の準備

「Google で登録する」をクリックします。



6.2. ニフクラ mobile backend の準備

自分の Gmail をクリックします。



6.2. ニフクラ mobile backend の準備

「メールアドレス」はそのままOKです。「確認メールを送信」をクリックします。



6.2. ニフクラ mobile backend の準備

確認メールが送られます。先ほど開いておいたGmailのタブを開きましょう。



6.2. ニフクラ mobile backend の準備

メールに書かれている「URL」をクリックします。

このメールは ニフクラ mobile backend から自動で送信しています。

ニフクラ mobile backend をご利用いただきありがとうございます。次のURLにアクセスすると、メールアドレスの確認が完了いたします。

▼URL

https://console.mbaas.nifcloud.com/signup/verify?token=_____



一定時間以内にご登録が完了しない場合は、上記のURLが無効となりますのでご注意下さい。

6.2. ニフクラ mobile backend の準備

「以上の規約に同意する」にチェックを入れて「アカウント登録」をクリックします。



6.2. ニフクラ mobile backend の準備

アカウントが作成され、ニフクラ mobile backend にログインできました🎉

The screenshot shows the 'New Application Creation' page of the NifCloud mobile backend interface. At the top, there is a navigation bar with links for 'アプリ一覧' (App List), 'ダッシュボード' (Dashboard), 'ドキュメント' (Documentation), '開発TIPS' (Development TIPS), 'コミュニティ' (Community), and '連携サービス' (Connected Services). On the right side of the header, there are icons for 'お知らせ' (Announcements) and a user account with the email 'natsumo2@gmail.com'. The main content area has a title 'アプリの新規作成' (New Application Creation). It contains a text input field labeled 'アプリ名' (App Name) with placeholder text '半角英数字もしくはアンダースコア' (Half-width English letters or underscores). Below the input field are two buttons: '戻る' (Back) and '新規作成' (Create New). At the bottom of the page, there is a footer with links for '富士通クラウドテクノロジーズホームへ' (Fujitsu Cloud Technologies Home), 'お問い合わせ' (Contact Us), '利用規約' (Terms of Use), '個人情報の取り扱いについて' (About Personal Information Handling), 'REST APIツール' (REST API Tools), and copyright information 'Copyright 2017 - 2019 FUJITSU CLOUD TECHNOLOGIES LIMITED'.

(参考) お家で mobile backend を使うときは、<https://mbaas.nifcloud.com/> からログインして使いましょう 🎉

6.2. ニフクラ mobile backend の準備

ニフクラ mobile backend 上にメッセージを保存するための場所を用意しましょう。
クラウドの中にもアプリを作ります。
「ProfileApp」と入力して「新規作成」をクリックします。



6.2. ニフクラ mobile backend の準備

クラウドにアプリが作成されました!

アプリが作成されると**APIキー**という2種類の鍵が発行されます。

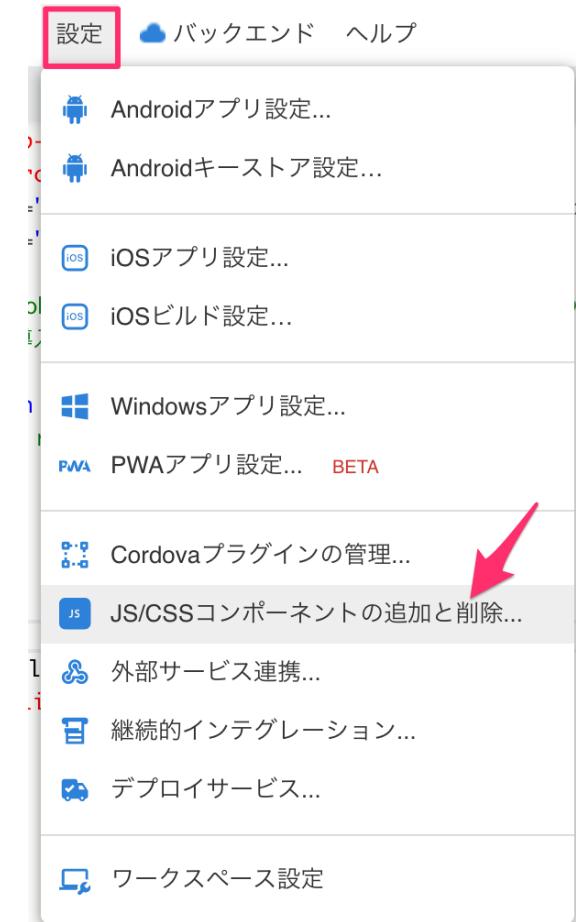


APIキーは後ほど使います。一旦このままにしておきましょう。

6.2. ニフクラ mobile backend の準備

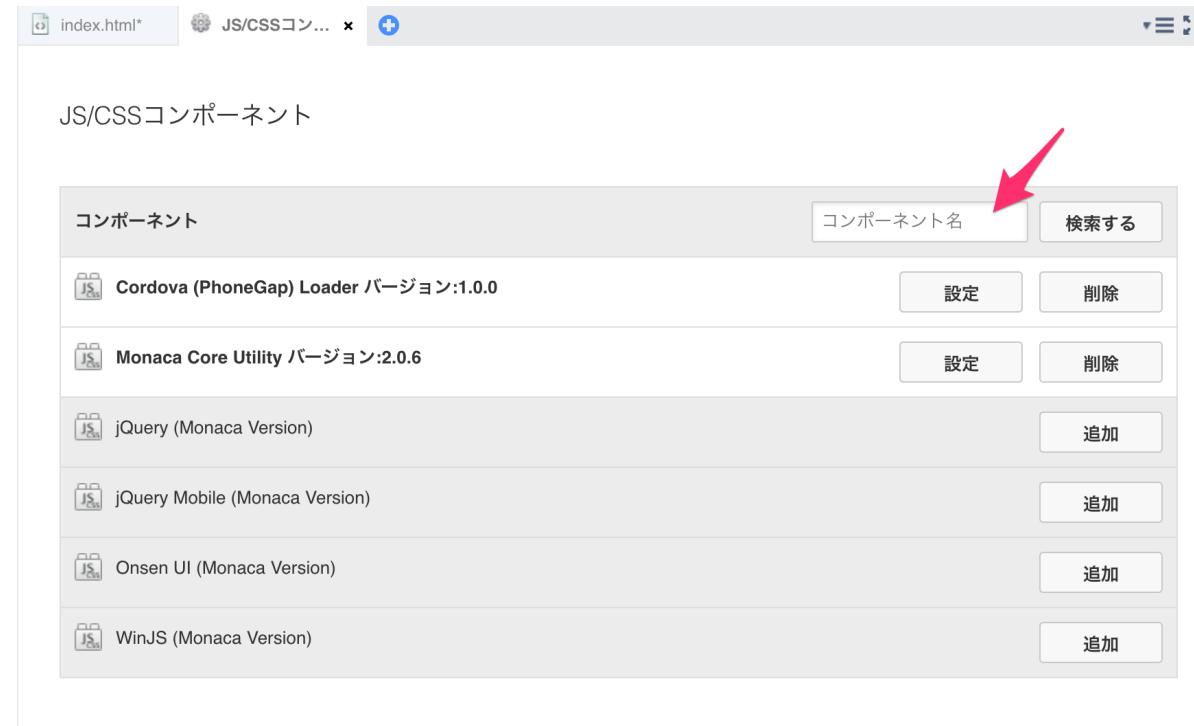
もう一度Monacaの開発画面のタブに戻りましょう。
クラウドの準備ができたので、
Monacaでクラウドを使う準備をしていきます。

「設定」をクリックして、その中にある
「JS/CSSコンポーネントの追加と削除」を
クリックします。



6.2. ニフクラ mobile backend の準備

「コンポーネント名」のところに「NCMB」と入力して、隣にある「検索する」をクリックします。



6.2. ニフクラ mobile backend の準備

「ncmb」が表示されるので「追加」をクリックします。

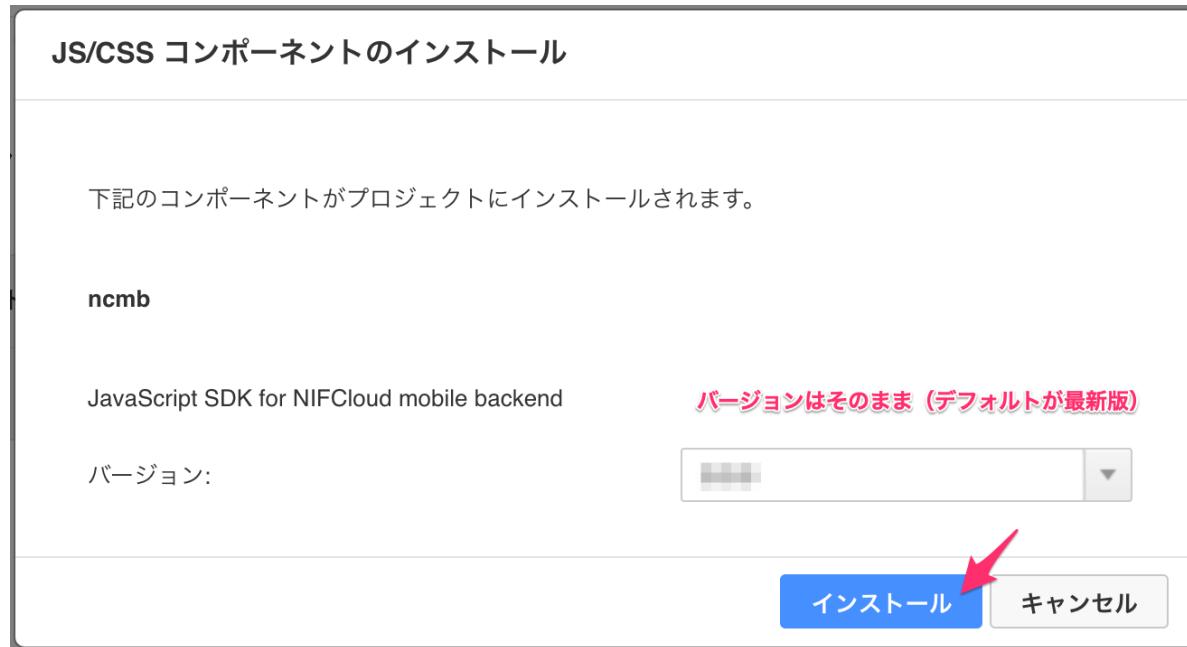


(参考) ニフクラmobile backend を Monaca で使う場合は、この「ncmb」を必ず追加する必要があるよ！

■ JavaScript SDK

6.2. ニフクラ mobile backend の準備

「インストール」をクリックします。



6.2. ニフクラ mobile backend の準備

必ずチェックを入れてから「OK」をクリックします。



6.2. ニフクラ mobile backend の準備

最初の一覧に戻ります。

一覧に「ncmb」が追加されていればOKです。

The screenshot shows the 'JS/CSSコンポーネント' (JS/CSS Components) list in the Monaca IDE. The search bar at the top right contains the text 'ncmb'. Below the search bar, there is a list of components:

- Cordova (PhoneGap) Loader バージョン:1.0.0
- Monaca Core Utility バージョン:2.0.6
- ncmb バージョン: [redacted]** (This row is highlighted with a red border)
- jQuery (Monaca Version)
- jQuery Mobile (Monaca Version)
- Onsen UI (Monaca Version)
- WinJS (Monaca Version)

Each component entry has '設定' (Settings) and '削除' (Delete) buttons on the right.

6.2. ニフクラ mobile backend の準備

この画面を閉じて、index.html を表示しましょう。



6.2. ニフクラ mobile backend の準備

index.html を編集します。index.html に元々ある**scriptタグ** (`<script>`, `</script>`) 内に次の内容をコピペします。

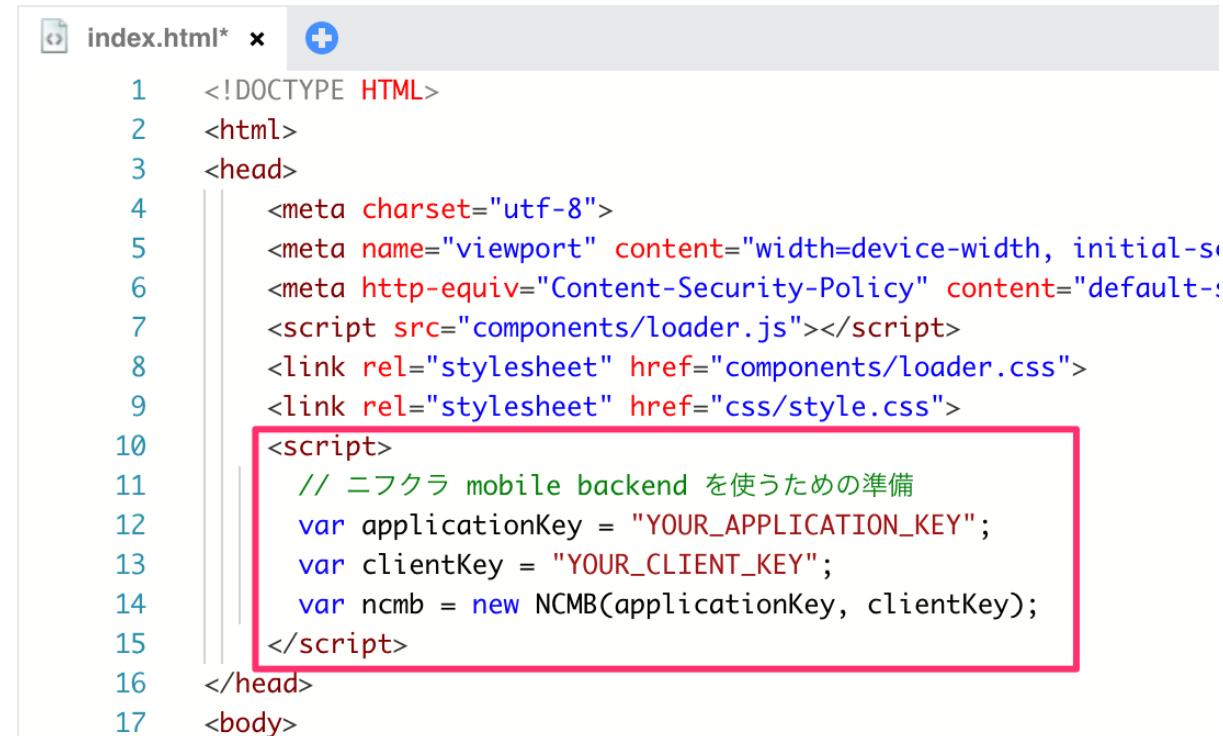
(次のコードをマウスで選択して、右クリックして「コピー」をクリックします。)

```
// ニフクラ mobile backend を使うための準備
var applicationKey = "YOUR_APPLICATION_KEY";
var clientKey = "YOUR_CLIENT_KEY";
var ncmb = new NCMB(applicationKey, clientKey);
```

SDKの初期化

6.2. ニフクラ mobile backend の準備

<script>と</script>の間で右クリックをして「ペースト」をクリックします。
次のように表示されればOKです。



```
index.html* x +  
1  <!DOCTYPE HTML>  
2  <html>  
3  <head>  
4      <meta charset="utf-8">  
5      <meta name="viewport" content="width=device-width, initial-s:  
6      <meta http-equiv="Content-Security-Policy" content="default:  
7      <script src="components/loader.js"></script>  
8      <link rel="stylesheet" href="components/loader.css">  
9      <link rel="stylesheet" href="css/style.css">  
10     <script>  
11         // ニフクラ mobile backend を使うための準備  
12         var applicationKey = "YOUR_APPLICATION_KEY";  
13         var clientKey = "YOUR_CLIENT_KEY";  
14         var ncmb = new NCMB(applicationKey, clientKey);  
15     </script>  
16 </head>  
17 <body>
```

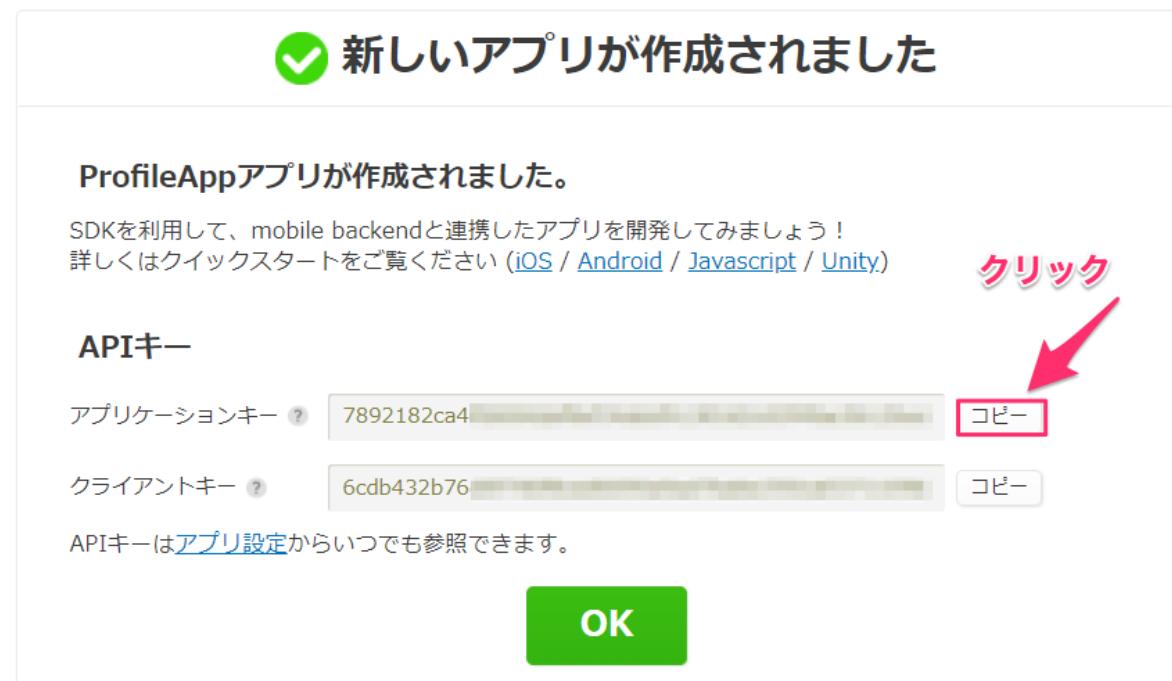
6.2. ニフクラ mobile backend の準備

`YOUR_APPLICATION_KEY`, `YOUR_CLIENT_KEY`の部分を書き換えます。

先ほど準備したニフクラ mobile backend のページを開きましょう。

```
// ニフクラ mobile backend を使うための準備
var applicationKey = "YOUR_APPLICATION_KEY";
var clientKey = "YOUR_CLIENT_KEY";
var ncmb = new NCMB(applicationKey, clientKey);
```

6.2. ニフクラ mobile backend の準備



6.2. ニフクラ mobile backend の準備

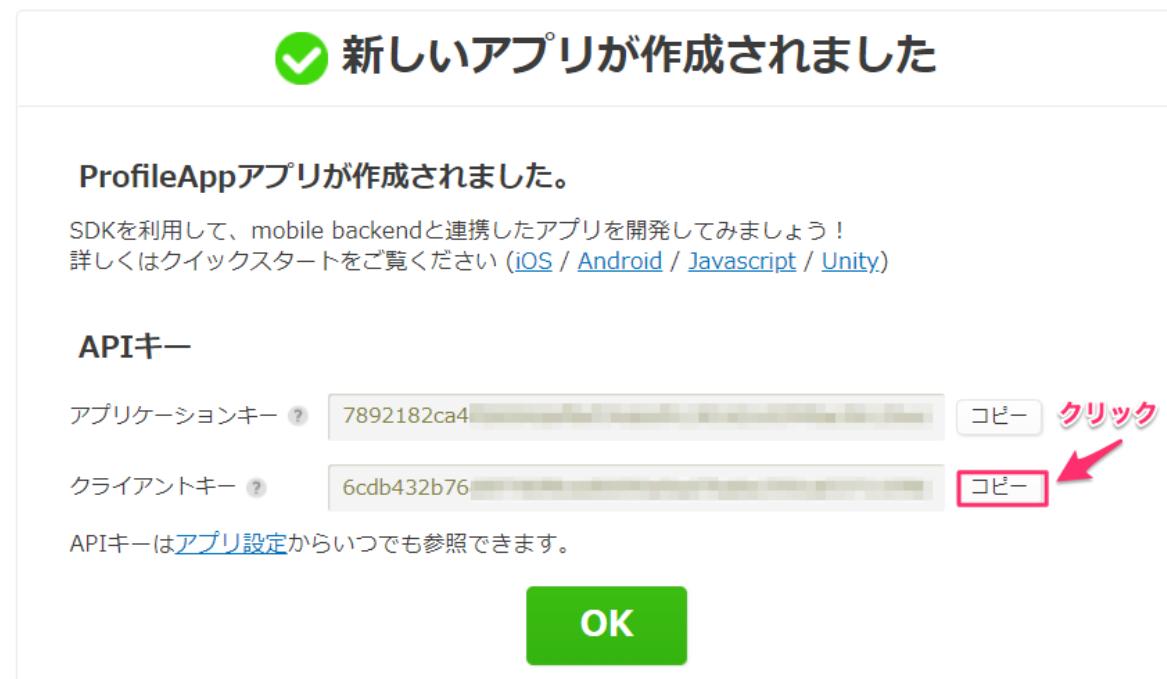
ダブルクリック→右クリックして
「貼り付け」をクリック

```
// ニフクラ mobile backend を使うための準備
var applicationKey = "YOUR_APPLICATION_KEY";
var clientKey = "YOUR_CLIENT_KEY";
var ncmb = new NCMB(applicationKey, clientKey);
```

6.2. ニフクラ mobile backend の準備

```
// ニフクラ mobile backend を使うための準備
var applicationKey = "7892182ca4f0[REDACTED]"; // 替え用
var clientKey = "YOUR_CLIENT_KEY";
var ncmb = new NCMB(applicationKey, clientKey);
```

6.2. ニフクラ mobile backend の準備



6.2. ニフクラ mobile backend の準備

ダブルクリック→右クリックして
「貼り付け」をクリック

```
// ニフクラ mobile backend を使うための準備
var applicationKey = "78921821f[REDACTED]";  
var clientKey = "YOUR_CLIENT_KEY";  
var ncmb = new NCMB(applicationKey, clientKey);
```

6.2. ニフクラ mobile backend の準備

```
// ニフクラ mobile backend を使うための準備
var applicationKey = "7892182ca4f████████████████████████";
var clientKey = "6cdb432b76d0████████████████████████";
var ncmb = new NCMB(applicationKey, clientKey);
```

これでクラウドの準備は完了です

6.3. コピペで機能追加

自己紹介を書いた bodyタグ の一番下に、次の内容をコピペします。

```
<!-- コメント入力機能 -->
<form name="messageForm">
  <input type="text" name="message" size="30" maxLength="20">
  <input type="button" value="送信" onclick="sendMessage();">
</form>
<!-- ログ表示 -->
<div id="log"></div>
```

6.3. コピペで機能追加

```
<body bgcolor="skyblue">
  <h1>私の<font color="red">ホームページ</font>アプリ</h1>
  <p align="center"></p>
  <p>私の名前は〇〇 〇〇です。小学校〇年生です。今学校では〇〇を勉強しています。</p>
  <h2>私の好きなもの</h2>
  <h3>食べ物</h3>
  <p>ハンバーグ、餃子</p>
  <h3>趣味</h3>
  <p>バスケ、算数</p>
  <!-- コメント入力機能 -->
  <form name="messageForm">
    <input type="text" name="message" size="30" maxlength="20">
    <input type="button" value="送信" onclick="sendMessage();">
  </form>
  <!-- ログ表示 -->
  <div id="log"></div>
</body>
```

これで画面にコメント入力欄が追加されますが、まだボタンを押した後の処理をコーディングしていないので、ボタンを押しても何も動きません。

6.3. コピペで機能追加

送信ボタンを押すとメッセージが保存されていくように処理を書いていきましょう

先ほど書いた内容に続けて、scriptタグ（`<script>`, `</script>`）内に次の内容をコピペします。

```
// メッセージが送信された時の処理
function sendMessage() {
    // 入力値
    var text = document.forms.messageForm.message.value;
    // 入力チェック
    if(text.length === 0) {
        // 画面に表示
        document.getElementById("log").innerText = "文字が入力されていません";
        // 処理中止
        return false;
    }
    // 保存先クラスを指定
    var MessageClass = ncmb.DataStore("MessageClass");
    // クラスのインスタンスを生成
    var messageClass = new MessageClass();
    // 値の設定と保存処理の実行
    messageClass.set("message", text)
        .save()
        .then(function(){
            /* 保存に成功した場合の処理 */
            // 画面に表示
            document.getElementById("log").innerText = "メッセージが保存されました!"
        })
        .catch(function(err){
            /* 保存に失敗した場合の処理 */
            // 画面に表示
            document.getElementById("log").innerText = "Error:" + err.code;
        });
}
```

6.3. コピペで機能追加

```
<script>
  var applicationKey = "fb81a2dac789e42bf[REDACTED]";
  var clientKey = "c1b4aedebba38349[REDACTED]";
  var ncmb = new NCMB(applicationKey, clientKey);

  // メッセージが送信された時の処理
  function sendMessage() {
    // 入力値
    var text = document.forms.messageForm.message.value;
    // 入力チェック
    if(text.length === 0) {
      // 画面に表示
      document.getElementById("log").innerText = "文字が入力されていません";
      // 処理中止
      return false;
    }
    // 保存先クラスを指定
    var MessageClass = ncmb.DataStore("MessageClass");
    // クラスのインスタンスを生成
    var messageClass = new MessageClass();
    // 値の設定と保存処理の実行
    messageClass.set("message", text)
      .save()
      .then(function(){
        /* 保存に成功した場合の処理 */
        // 画面に表示
        document.getElementById("log").innerText = "メッセージが保存されました！"
      })
      .catch(function(err){
        /* 保存に失敗した場合の処理 */
        // 画面に表示
        document.getElementById("log").innerText = "Error:" + err.code;
      });
  }
</script>
```

6.4. 動作確認



6.4. 動作確認



6.4. 動作確認



6.4. 動作確認

The screenshot shows the mobile backend dashboard interface. On the left, there's a sidebar with icons for 'メニューを閉じる', 'ダッシュボード', '会員管理' (highlighted with a red arrow), 'データストア' (highlighted with a red box), 'ファイルストア', 'スクリプト', and 'プッシュ通知'. The main area is titled 'ProfileApp' and shows the 'MessageClass' data store. The 'MessageClass' tab is selected. The table displays a single record:

objectId	message	createDate	updateDate
5ScHsFQbogK4ahQ7	私もハンバーグが好きです！	2019-08-02T00:31:22.205+09:00	2019-08-02T00:31:22.206+09:00

A red arrow points to the message content '私もハンバーグが好きです！' with the text 'メッセージが保存されていることを確認できます'.

7. アプリを使って自己紹介しよう！

自己紹介手順

まとめ

まとめ

今日できたこと！

まとめ

まとめ

参考書の紹介

