

# アプリづくりの準備



ここからは「Monaca (モナカ)」という、スマホアプリをつくるためのWebサイトを使うよ。Monacaを使えば、特別なアプリをインストールしなくてもWebブラウザーだけでアプリを開発することができるんだ。つくっているアプリを試しに動かしたりも簡単にできて便利だよ。Monacaへアクセスするときには、Part 4でも使ったChromeを使うようにしよう。ほかのWebブラウザーだとうまく動作しないことがあるから注意してね。Macを使っている場合は、Mac用のChromeでアクセスすれば使い方はいっしょだよ。

1 Monaca のアカウント登録

Part 5 の内容を行うには、Monaca のサイトで無料登録をして、アカウントを作成する必要があります。プログラミングサポートページの「Monaca のアカウント登録」の項目で紹介している手順で、大人といっしょに必要な手続きを行ってください。

<https://kodomonokagaku.com/miraiscience/support/>

1 ダッシュボードを開く

Monaca で作業をはじめるためには、Dashboard (ダッシュボード) というページを使うよ①。このダッシュボードのページは、つくっているアプリの一覧を見ることができ、選択するとアプリの内容を編集することができる。Scratch でいうところの、「私の作品」のページと同じだね。



## 2 Monaca にログイン

ダッシュボードのページを見るためには、Monaca のサイトにログインしておく必要がある。一度作業をやめて、Monaca のサイトにログインしていないときは、ログインしてから作業をはじめよう。画面右上にあるログインを選択して②、メールアドレスとパスワードを入力して、ログインのボタンをクリックしよう③。これもScratch を使う場合といっしょだから大丈夫だよね。

2 Monaca にログイン



## 3 クラウド IDE

Monaca の無料 (Free) プランを使う場合は、アドリのプロジェクト (Scratch のプロジェクトと同じだ) は 3 つまでしかつくれないという制限がある。アカウント作成をした段階で、はじめてから「はじめての Monaca アプリ」というサンプルのプロジェクトが用意されているから、まずはそれを開こう。プロジェクトの一覧からプロジェクトを選択して④、「クラウド IDE を開く」というボタンをクリックしよう⑤。

3 クラウド IDE

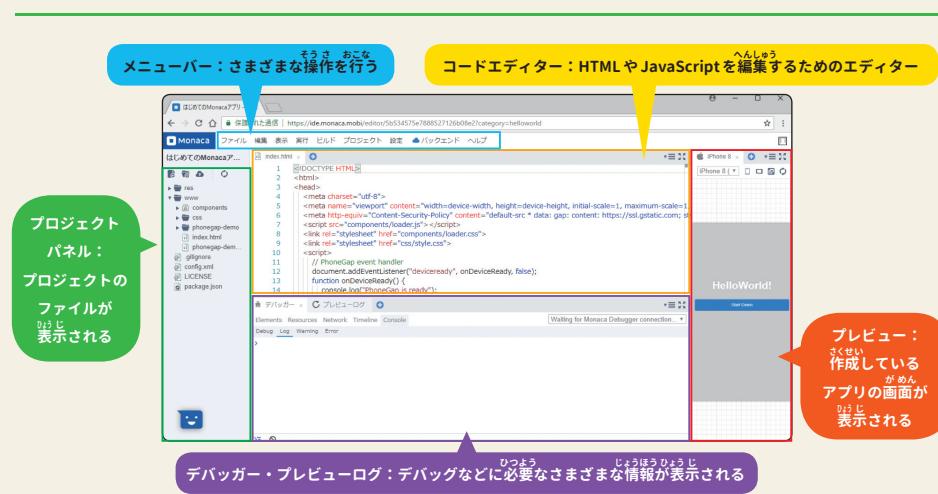
4 はじめてのMonacaアプリ

5 クラウド開発

6 の画面が開いたかな？ Monaca では、このクラウド IDE と呼ばれる画面を使ってアプリをつくっていくよ。



クラウド IDE  
の使い方は  
次のページへ！



IDE というの、統合開発環境 (Integrated Development Environment) の略で、プログラムを書くためのエディターだけでなく、バグの発見に役立つデバッガーというツールや、つくったアプリをスマホに転送するためのツールなど、プログラミングをするときに必要になるいろいろなツールをまとめて、使いやすくしたもののことだよ。

画面の真ん中に表示されているのがコードエディターだ。ここに

にゅうりょく  
HTML や JavaScript などを入力していくよ。

画面の左に表示されているのは、プロジェクトパネルで、ここは

つかひょうじいわく  
アプリで使っているさまざまなファイルが表示されている。この一

から HTML や JavaScript を書いたファイルをダブルクリックすると、**⑦**、コードエディターで開いて編集ができる。

画面の右に表示されているのはプレビューだ。つくっているアプリの見た目がどのよ

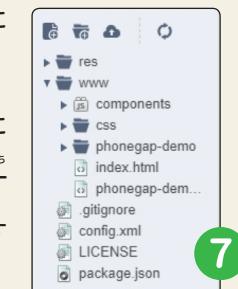
うになっているかを表示している。Part 4 の 145 ページで紹介した Brackets のライ

Preview と同じだと思えばいい。画面の下にあるのはデバッガー・プレビューログだ。

プログラムのエラーや、アプリの動作の情報などを確認することができるよ。

画面の右上には、人型のアイコンが表示されているね。ここを  
が めん もど

クリックすると、タッシュボードの画面に戻ったり、Monacaのサイトからログアウトしたりすることができるぞ⑧。



1



8

# あいさつアプリ をつくろう



最初はシンプルなアプリをつくりながら、Monaca の使い方や基本的なアプリのつくり方を解説するよ。最初につくるのは「あいさつを表示してくれるアプリ」だ。



あたら  
さっそく新しくプロジェクトをつくってみ  
よう。MonacaのプロジェクトはScratchの  
おな  
プロジェクトと同じで、これからつくるアプ  
かんり  
リを管理するための入れ物のことだよ。

が めん ひだりうえ あたら  
ダッシュボードの画面の左上にある「新  
しいプロジェクトを作る」というボタンを  
クリックしよう❶。プロジェクトのテンプ  
レートを選択する画面が表示される。テン  
プレートは、アプリをつくるために必要な  
ファイルの「ひな形」のことだよ。つくり  
たいアプリに応じて、いくつかの種類が用  
意されているんだ。今回はシンプルなアプ  
リをつくる練習をしたいから、一番右の

「最小限のテンプレート」で「作成」を選択しよう③。プロジェクト名を入力する画面が表示されるので、「あいさつアプリ」と入力して④、「プロジェクトを作成する」を選択しよう⑤。今回は説明については入力しなくてOKだ。



156



## 2 IDEを開く

プロジェクトの一覧に「あいさつアプリ」が追加されたね。「クラウドIDEで開く」を選択してIDEの画面に切り替えよう⑥。「あいさつアプリ」を編集できるIDEの画面が開いたね⑦。

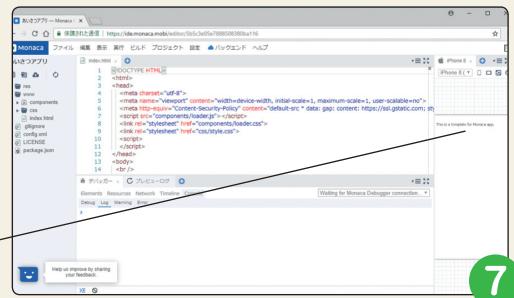
プレビューには、「This is a template for Monaca app. (これは Monaca でつくるアプリのひな形です)」と表示されているはずだ。

プレビューの  
最初の表示

This is a template for Monaca app.



6



7



## 3 HTMLの編集

コードエディターでindex.htmlが開いていることを確認して、次の文字列が書かれている部分を探してみよう。

14行目に改行のための「  
」のタグがあり、その下の15行目に「This is a template for Monaca app.」という文字列があるね。これを「あいさつアプリ」という一番大きな表題に書き換えてみよう⑧。表題を書くためのタグは覚えているかな?「h1」のタグだったね。Monacaのコードエディターも、Bracketsと同じように、タグの候補を自動的に表示してくれる機能(141ページ参照)があるから活用しよう。

14行目と15行目を削除

あいさつアプリという表題を  
表示するように変更

8

```
13 <body>
14 <br />
15 || This is a template for Monaca app.
16 </body>
```

```
13 <body>
14 | <h1>あいさつアプリ</h1>
15 </body>
```



## 4 保存する

うまく修正できたら、ファイルメニューから、「保存」を選択⑨して、編集したindex.htmlを保存しよう。

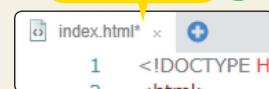
編集をして、保存をしていないファイルについては、エディターの上にあるファイル名の最後に「\*」のマークが表示されるよ⑩。保存するとこのマークは消えるぞ。

保存をすると自動的にプレビューが更新されて、表示されている文字が「あいさつアプリ」に変わったはずだ⑪。ここまでPart 4でやったHTMLの編集とほとんど同じだから簡単だよね。

もし、プレビューが自動的に更新されない場合、更新のためのボタン(丸い矢印)を押せば⑫、最新の内容に更新されるよ。



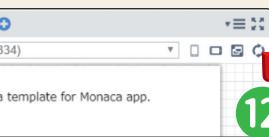
9



10



11



12



## 5 JavaScriptとは?

13

```
10 <link rel="stylesheet" href="css/style.css" />
11 <script>
12 </script>
</head>
```

Scriptのタグの中に  
JavaScriptを書いていく

次はJavaScriptを書いてみよう。

JavaScriptはWebブラウザで実行することができるプログラミング言語だ。JavaScriptを使えば、いろいろなしきみをもったWebサイトをつくることができる。今回はWebサイトではなく、スマホアプリをつくっているけれど、スマホのOSには、HTMLを表示したり、JavaScriptを実行したりするための部品(WebViewと呼ばれる)が内蔵されている。HTMLで画面をつくり、JavaScriptを使ってしきみをつければ、スマホのアプリとして動作させることができるんだ。HTMLファイルの中に「script」というタグを書いて、その中にJavaScriptを書くことができるぞ⑬。



## 6 JavaScript を書こう

index.html の 10 行目と 11 行目を見てみよう。じつは「script」タグはすでに用意されているぞ。この script タグの中に JavaScript を書けば、index.html が表示されたときに実行されるよ。

あいさつアプリという名前にしたけれど、まだ何のあいさつもしてくれないよね。「アラート」という小さなウインドウを表示させて、そこにあいさつを表示させるようにしてみよう。アラートを表示させて、その中にメッセージを表示させるための JavaScript は⑯のようになるぞ。「こんにちは」以外はすべて半角で入力することに注意しよう。

Scratch の「と言う」のブロック⑯と同じだと考えるとわかりやすいね。Brackets と同じように、自動的にインデント (141 ページ参照) がついて、script の開始タグ (10 行目) の最後で改行を入力すれば、インデントされた状態で次の行にカーソルが移動するぞ。

```
10  <script>
11  alert("こんにちは");
12  </script>
```

インデント  
ダブルクオーテーションで囲む  
カッコを対応させる  
最後はセミコロンを  
わざ  
忘れない

15 こんにちは と言う

Scratch の  
ブロックだと…

## 7 保存する

入力できたら、保存をしよう。保存すると、⑯のようなアラートが表示される。もし表示されない場合は、JavaScript が間違っていないかを確認してみよう。少しあわてにくいけれど、これは Monaca のプレビューで、今書いたアラートが表示されたということだよ。



16



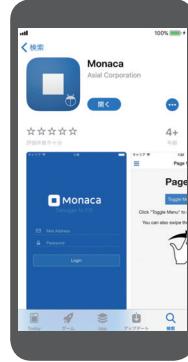
## 8 スマホで動作確認しよう

つくったあいさつアプリをスマホで動かしてみよう！ Monaca には「実機デバッグ」という機能があって、つくったアプリをスマホで試しに動かしてみることができるよ。



### Monaca デバッガーのインストール

つくったスマホアプリを実際のスマホで動かすためには、7 ページにある環境を備えたスマホの機種を用意した上で、「Monaca デバッガー」という無料のアプリをインストールする必要があります (iPhone、Android 対応)。以下のプログラミングサポートページの「Monaca デバッガー」のインストールの項目で紹介している手順でアプリをインストールして、Monaca と同じメールアドレスとパスワードでアプリにログインしてください。

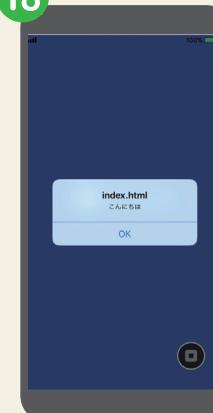


<https://kodomonokagaku.com/miraiscience/support/>

18



17



18



19

画面の右下に表示されているのは、デバッガーメニューを表示するためのボタンだよ⑯。これを押すと、メニューが表示されて、いろいろな操作ができるぞ⑰。

Monaca デバッガーのアプリにログインすると、Monaca.io プロジェクトの欄に今つくっている「あいさつアプリ」が見えるはず。これをタップする⑯。

20



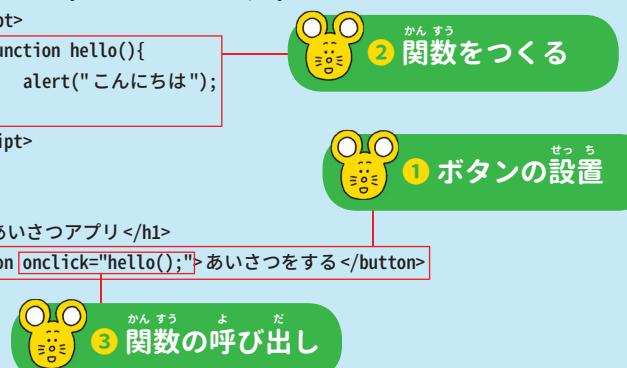
プロジェクトの一覧に戻る  
プロジェクトを最新の状態に更新する  
スクリーンショットを撮る  
アプリの動作ログを見る

# あいさつアプリを かいぞう 改造しよう



ここまでつくったあいさつアプリは、アプリを起動するとすぐにアラートが表示されていたね。「イベント」と「関数」を使って、アプリを起動した後で、ボタンを押すとあいさつを表示するように改造してみよう。まずは下に、次の①から③で作成・編集するHTMLやJavaScriptを整理しておくよ。どこを書いているかわからなくなったら、このページに戻って確認しよう。

```
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,
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         function hello(){
12             alert("こんにちは");
13         }
14     </script>
15 </head>
16 <body>
17     <h1>あいさつアプリ</h1>
18     <button onclick="hello();">あいさつをする</button>
19 </body>
20 </html>
```



## 1 ボタンの設置

「イベント」というのは、ユーザーがボタンをクリックしたり、ページを移動したり、ユーザーの操作によって生じる「できごと」のことだよ。ここでは、ボタンを押したらあいさつを表示するようにアプリを改造するぞ。

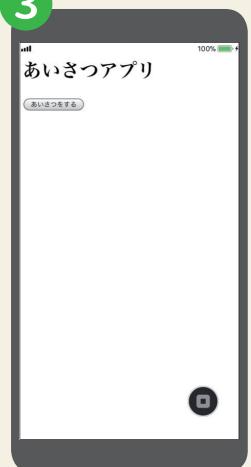
14 <body>
15 <h1>あいさつアプリ</h1>
16 <button>あいさつをする</button>
17 </body>

## あいさつアプリ

あいさつをする

まずはボタンを表示させるためのHTMLをindex.htmlに加えよう。ボタンを表示させるためには、「button」というタグを使うよ。お手本を参考にして追加してみよう①。buttonというタグを使うと、HTMLでクリックできるボタンを表示させることができる。タグで囲んだ内容はボタンのラベルになるよ。今回は「あいさつをする」というラベルにしてみた。HTMLが完成したら、保存をして、プレビューでボタンが表示されたか確認してみよう②。

スマホでMonacaデバッガーを表示させたままにしておくと、更新を自動で検知して、表示を更新してくれるぞ③。



ボタンが  
ついたよ！





## 2 関数をつくる(定義する)

さて、今まではボタンを押してもな  
にもおこらないし、index.htmlを保存す

るとアラートが表示されてしまうね。これを改善してみよう。

まずは、JavaScriptのアラートを表示している部分を「関数」として書き直す必要がある。関数というのは、プログラムの処理をまとめて、名前をつけたものだよ。今回は、アラートを表示する処理をまとめて、helloという名前をつけた関数をつくるみよう

### ④ 関数のつくり方の基本は右の図の

通りだ。関数をつくることを、関数を「定義する」ともいうよ。関数の名前は半角英数でつけることに注意しよう。

```
10  <script>
11  function hello(){
12  alert("こんにちは");
13  }
14  </script>
```

4

カッコを忘れないように



## 3 関数の呼び出し

ダブルクオーテーションで囲む

```
16  <body>
17  <h1>あいさつアプリ</h1>
18  <button onclick="hello();">あいさつをする</button>
19  </body>
```

helloという名前の関数がつくれたら、ファイルを保存しよう。ファイルを保存してもアラートは表示されなくなったね。関数をつくっただけでは、関数としてまとめた処理は実行されないんだ。関数の名前を指定して「呼び出し」をすると、関数としてまとめた処理が実行されるようになるぞ。

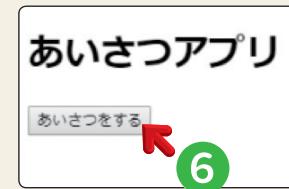
今回は、ボタンが押されたらこのhelloという関数を呼び出すようにしてみる⑤。ボタンのHTMLにクリックされたイベントを調べて、helloという関数を呼び出すよな  
しくみを追加しよう。ここでは、ボタンのHTMLにonclickという部分を追加して、helloという名前の関数を呼び出すようにしているよ。



## 4 クリックして確認

保存をしたら、プレビューに表示されているボタンをクリックして⑥、アラートが表示されることを確認しよう⑦。

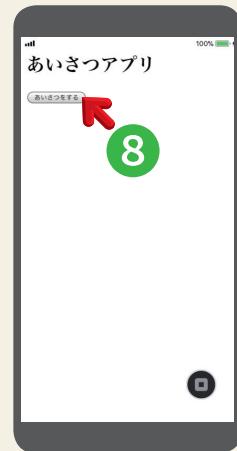
Monaca デバッガーでも同じように動作をすることを確認してみよう⑧⑨。



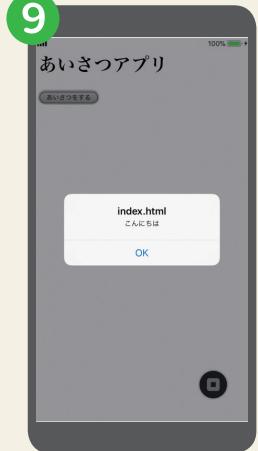
6



7

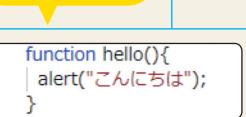
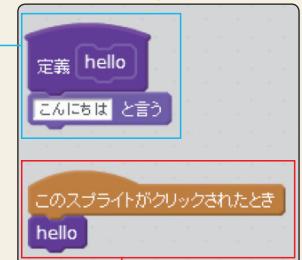


8



9

Scratchのブロックだと…



じつはこの関数に似たものを作る  
機能は Scratch にも用意されている

よ。Scratch では「その他」のカテゴリから、新しい  
ブロック自分でつくることができる。たとえば、  
helloというブロックをつくる、内容を「こんにちは」という吹き出しが表示されるようにしておく。  
スプライトがクリックされたら、helloというブ  
ロックを実行するようにしたのと  
同じことだよ。少し違うのは、今  
回つくったHTMLでは、ボタンと  
いう別の部品(スプライト)がク  
リックされたらhelloという関数  
(ブロック)を実行できるように  
したことだね。



## 5 入力の受け取り

```

10 <script>
11 function hello(){
12     var name = prompt("お名前は？");
13     alert(name + "さん、こんにちちは");
14 }
15 </script>

```

最後の仕上げに、162ページのHTMLとJavaScriptからさらに、自分の名前を入力して、あいさつに含めるようにしてみよう。⑩のように書いていくよ。完成した動作は⑪のようになる。Monaca デバッガーでも、名前を入力してあいさつを表示させることができるぞ。

では、書いたプログラムのポイントを解説していくよ。ユーザーからの入力を受け取るときは、「prompt」という命令を使うよ。そして、JavaScriptでも Scratchと同じように変数を使うことができるぞ。

12行目は入力された名前を保存しておく部分だね。ここでは名前を保存しておくためにnameという変数を用意した。JavaScriptの場合は、

10

11



はんかく 半角のスペース  
変数 name に入力された文字列を代入  
var name = prompt("お名前は？");  
ひょうじ 入力用のボックス (プロンプト) を表示  
なまえ 最後はセミコロンを わず 忘れない

へんすう もじ 変数と文字をつなげる  
alert(name + "さん、こんにちちは");  
へんすう ないよう 変数nameの内容と ひょうじ 「さん、こんにちちは」をつなげて表示する

var の後に半角スペースを入れて、変数の名前を書くよ。Scratch の場合は、「変数を作る」というボタンを押して、変数の名前を入力したよね。JavaScript の場合は変数の準備についても文字で書くぞ。

Scratch と違って、変数の名前には漢字やひらがなは使えないから注意しよう。変数に代入する場

合は=(イコール)を使う。この場合は、ユーザーが入力用のボックス(プロンプト)に入力した文字列(名前)を代入している。Scratchの場合は「=」の記号は代入ではなく、比較をするときに使っていたから注意しよう。

13行目では、変数 name には、入力された名前が代入されているから、これと「さん、こんにちちは」という文字列をつなげて表示している。変数と文字列をつなげるために+(プラス)の記号を使うよ。

Scratch にも入力用のボックスを表示するため、「と聞いて待つ」というブロックが用意されているから、ほとんど同じことができる。参考のために Scratch で同じことをした場合のスクリプトも載せておくね⑫。Scratch の場合は、入力された文字列は「答え」という変数に代入されるようになっているのがJavaScriptと違うところだね。

name という名前の変数を用意して、杉浦という文字列を代入する

JavaScriptでは…

var name = "杉浦";



12 Scratchのブロックだと…



※ 関数や変数に名前をつけるときは、次の3つのこと注意しよう。

① 数字、英字、\_と\$の記号が使える

② JavaScriptの予約語(特別な意味を持つ単語)を名前にすることはできない

break・case・catch・continue・default・delete・do・else・false・finally・for・function・if・in・instanceof・

new・null・return・switch・this・throw・true・try・typeof・var・void・while・with

③ 数字から始まる名前をつけることはできない

# クイズアプリをつくろう



ここからは今まで解説したことも使って、プログラミングに関するクイズのアプリをつくってみよう。ぜんぶで4つの画面のアプリだ。



## 1 クイズアプリの設計

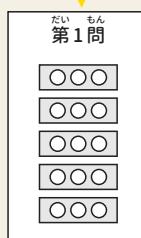
まずはこれからつくるアプリの内容を整理しておこう①。アプリを起動すると、クイズが3問出題される。最初の問題はボタンで答える5択の問題だ。次は絵をタップして答える3択の問題。最後は、答えを文字で入力をして答えるタイプの問題だよ。

3問の問題に答え終わると、最後のページで正解数が表示され、正解数によってコメントが変わるようにしてみよう。

ここではプログラミングに関するクイズを例にしたけど、問題のテーマはなんでもいいよ！やり方がわかったらオリジナルのクイズアプリをつくってみよう。



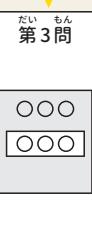
1 ボタンをクリックして答える



2 画像をクリックして答える



3 入力をして答える



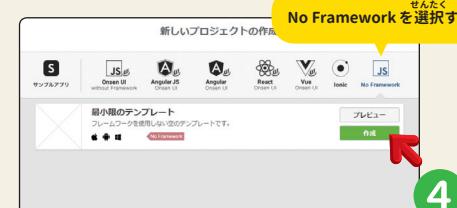
4 正解数でコメントが変化する



## 2 準備しよう



あたら「新しいプロジェクトを作る」をクリックする②。



あいさつアプリと同じように、No Frameworkの最小限のテンプレートを選択して③、作成をクリックしよう④。



名前はクイズアプリにして、説明もそえてプロジェクトを作成しよう⑤。



プロジェクトがつくれたら、クラウドIDEで開くをクリックしてIDEを起動するよ⑥。



最初にクイズアプリに必要なHTMLのファイルをぜんぶ用意しておこう。画面左のプロジェクトパネルのindex.htmlを右クリックし、「ファイルをコピー」を選択しよう⑦。



ファイル名にはquiz2.html(2問目を出題するページ)と入力して⑧、OKをクリックしよう⑨。



同じようにindex.htmlをコピーする作業をしてquiz3.html(3問目を出題するページ)とrecord.html(結果を表示するページ)をつくろう⑩。



### ③ HTMLの編集と正解判定

まずは1ページ目(index.html)からつくっていくよ。HTMLを編集して、問題と答えを選ぶボタンを表示させよう⑪。

これはこれまでの復習だから簡単だね。

次にそれぞれのボタンを押したときに実行される関数をつくるよ。正解のボタンは5つのうち1つだけだ。正解だったときに実行される関数をcorrect(正解)、間違いだったときに実行される関数をincorrect(不正解)という名前でつくるよ⑫。

関数の内容として、正解か不正解かを知らせるためのアラートを表示するようにしておこう。ちなみに、正解(日本で作られたプログラミング言語)はRubyだ。

次にHTMLのボタンの部分に⑫でつくった関数の呼び出しを加えるよ⑬。Rubyのボタンだけはcorrect関数を呼び出すようにして、それ以外のボタンはみんなincorrect関数を呼び出すようにすればいいね。この段階でボタンを押してみて⑭、正解(すごい！正解だよ！)か不正解(残念！はずれ！)のアラートが表示されることを確認しておこう⑮。

この画面をつくるよ！

#### 第1問

日本で作られたプログラミング言語は？

```
13 <body>
14 <h1>第1問</h1>
15 <p>日本で作られたプログラミング言語は？</p>
16 <button>Scratch</button>
17 <button>Python</button>
18 <button>Ruby</button>
19 <button>Swift</button>
20 <button>Java</button>
21 </body>
22 </html>
```

11

```
9 <link rel="stylesheet" href="css/style.css">
10 <script>
11 function correct(){
12   alert("すごい！正解だよ！");
13 }
14
15 function incorrect(){
16   alert("残念！はずれ！");
17 }
18 </script>
19 </head>
20 <body>
21 <h1>第1問</h1>
22 <p>日本で作られたプログラミング言語は？</p>
```

12

Rubyのボタンだけcorrect関数を呼び出すようにする

```
19 </head>
20 <body>
21 <h1>第1問</h1>
22 <p>日本で作られたプログラミング言語は？</p>
23 <button onclick="incorrect();">Scratch</button>
24 <button onclick="incorrect();">Python</button>
25 <button onclick="correct();">Ruby</button>
26 <button onclick="incorrect();">Swift</button>
27 <button onclick="incorrect();">Java</button>
28 </body>
29 </html>
```

13

#### 第1問

日本で作られたプログラミング言語は？

...144b235d365ab74e5c70be--507299520-8080.monaca-app.comに埋め込まれているページの内容  
すごい！正解だよ！

OK

15

Monaca デバッガーでも同じようにボタンをタップして⑯、表示がでることを確認しておこう⑰。

### ④ ページを移動する処理

次に、ページを移動するための処理を追加するよ。今つくっているのは最初の問題を出題するためのページだね。次のページはquiz2.htmlというファイル名でつくってある。正解のときも不正解のときも、次の

quiz2.htmlに移動する

ように処理を追加しよう。location.hrefに移動したいページの名

前を代入すると、ページを移動することができる⑯。

これも実際にボタンを押して、動作確認をしておくとよいね⑯。まだquiz2.htmlのページは編集をしていないから、ボタンを押して、「This is a template for Monaca app.」と書かれたquiz2.htmlのページが表示されればOKだ。

プレビューの表示をindex.htmlに戻したいときは、更新ボタンを押せばいいぞ⑯。Monaca デバッガーの場合は、デバッガーメニューから更新ボタンをタップしよう⑯。



```
9 <link rel="stylesheet" href="css/style.css">
10 <script>
11 function correct(){
12   alert("すごい！正解だよ！");
13   location.href = "quiz2.html";
14 }
15
16 function incorrect(){
17   alert("残念！はずれ！");
18   location.href = "quiz2.html";
19 }
20 </script>
21 </head>
22 <body>
```



21

## 5 正解数のスコア記録

1 ページ目の製作の仕上げとして、正解数のスコアを記録する方法について解説するね。score という変数をつくって、そこに正解数を保存すればよいのではないかと思うよね？ 基本的な考え方はその通りだけれど、このアプリではぜんぶで 4 ページの HTML を用意して、順番に表示していく。たとえば index.html のページで score という変数を用意して、score に数字を代入しても、次の quiz2.html のページではこの score に代入されている値を参照する（読み出す）ことはできない②。

そこで、「セッションストレージ」というしくみを使って正解数のスコアを保存することにするよ。このしくみを使うと、開いたページをまたいで変数を使うことができるぞ。使い方は変数とほとんど同じだけど、少しだけ書き方が違うよ。ここではセッションストレージの中に score という変数をつくって、そこに正解数を保存するよ③。

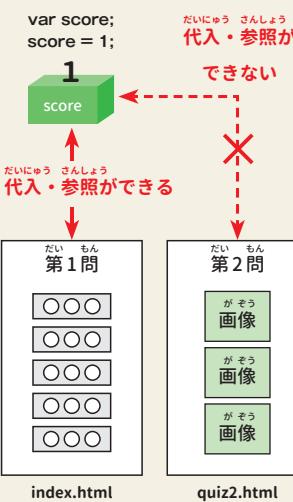
正解の場合は 1 を、不正解の場合は 0 を保存しておこう④。

```

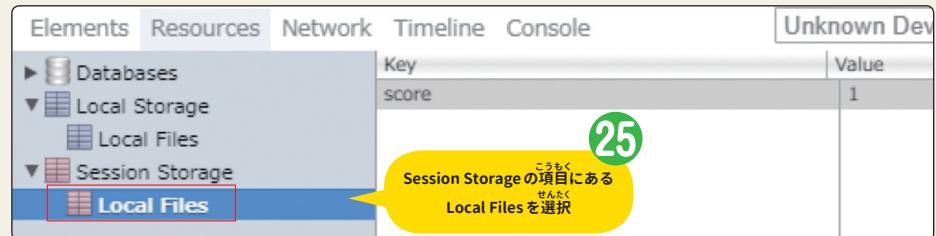
9  <link rel="stylesheet" href="css/style.css">
10 <script>
11   function correct(){
12     alert("すごい！正解だよ！");
13     location.href = "quiz2.html";
14     sessionStorage.score = 1;
15   }
16
17   function incorrect(){
18     alert("残念！はずれ！");
19     location.href = "quiz2.html";
20     sessionStorage.score = 0;
21   }
22 </script>

```

22



Monaca デバッガーを使っているときは、デバッガーの Resources というタブからセッションストレージの内容を確認することができるぞ⑤。



さいしょ 最初のページ (index.html) の HTML と JavaScript を⑥にまとめておくね。

26

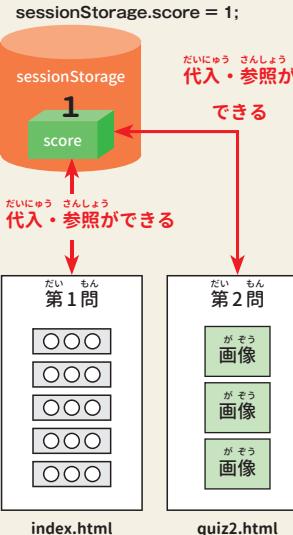
```

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,
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     function correct(){
12       alert("すごい！正解だよ！");
13       location.href = "quiz2.html";
14       sessionStorage.score = 1;
15     }
16
17     function incorrect(){
18       alert("残念！はずれ！");
19       location.href = "quiz2.html";
20       sessionStorage.score = 0;
21     }
22   </script>
23 </head>
24 <body>
25   <h1> 第1問 </h1>
26   <p> 日本で作られたプログラミング言語は？ </p>
27   <button onclick="incorrect();>Scratch</button>
28   <button onclick="incorrect();>Python</button>
29   <button onclick="correct();>Ruby</button>
30   <button onclick="incorrect();>Swift</button>
31   <button onclick="incorrect();>Java</button>
32 </body>
33 </html>

```

ついでに  
追加した  
HTML

23





## 6 画像の準備

次は2ページ目(quiz2.html)をつ  
くっていこう。まずはクイズの画像を  
用意しておこう。今回はScratchに登  
場する3つのオリジナルキャラクター  
の名前(ギガ、ナノ、ピコ)を当てるク  
イズにしたよ。用意する画像のファイル  
名は半角英数字でつけておこう②7。  
これらの画像の保存場所はデスクトッ  
プにしておこう。

この画像ファイルをMonacaで使う  
には、ファイルをアップロードする必  
要がある。IDEの画面の左にあるプロ  
ジェクトパネルから、wwwという  
フォルダーのアイコンを右クリックし  
てメニューを表示させ、「ファイルを  
アップロード」を選択しよう②8。表示  
されたウインドウの真ん中の部分に、  
デスクトップにある画像ファイルをド  
ラッグするか、ファイル選択ボタンを  
押して画像ファイルを選択して、3枚  
ともアップロードしよう②9。

アップロードが終わったら、×

プロジェクトパネルに3つのファイルが追加されているか確認しよう②10。ファイルを  
ダブルクリックすれば、画像を見ることもできるよ②11。

27



28



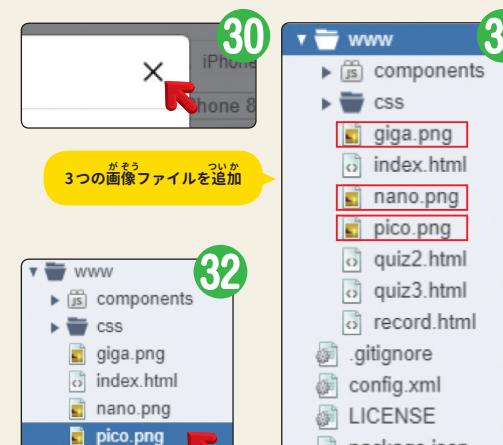
29



30



31



32



## 7 問題文と画像表示のタグ

12 </head>

13 <body>

14 <h1>第2問</h1>

15 <p>Scratchのキャラクター、ピコは? </p>

16 

17 

18 

19 </body>

20 </html>

33

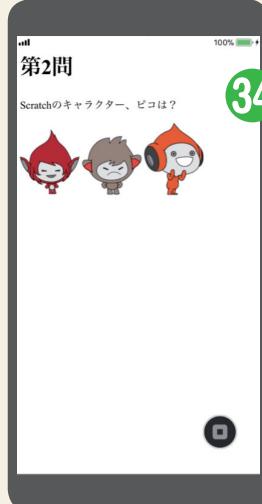
ファイルがアップロードできたら、問題文と画像を表示するためのタグをquiz2.htmlに書き加えよう。まずはquiz2.htmlをダブルクリックして、エディターを開き、3つのimgタグを加えるよ③3。この画像の表示の方法はPart 4で解説した内容だから、大丈夫だね。

プレビューでは、常にindex.htmlの内容が表示されるようになっている。今つくっているquiz2.htmlの画面を確認したい場合は、第1問に回答をして、次の第2問に進めばquiz2.htmlの内容を確認することができるぞ。

スマホは画面が小さいので、一列に3つの画像が表示されるように、画像の横幅を90px(ピクセル)に指定したよ。widthの前に半角でスペースを忘れないように注意しよう。使っているスマホの画面が狭い場合は、90pxという数値を変える必要がある。実際にMonacaデバッガーで確認してみるといいぞ③4。

第2問

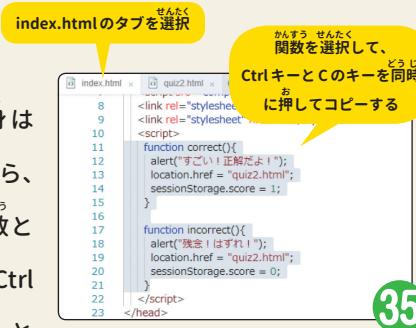
Scratchのキャラクター、ピコは?



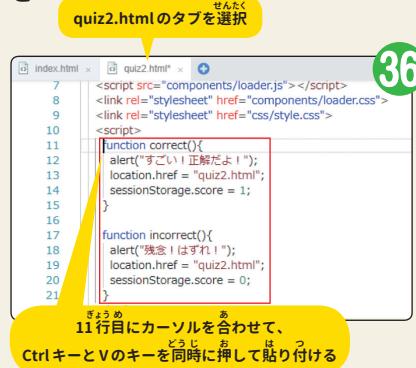
34



## かん すう 閑数をコピーする



35



36



## 2ページ目の仕上げ



Scratchの  
ブロックだと…

2つ目は、スコアの計算。正解の場合は、第  
もん かくとく だい もん かくとく  
1問で獲得したスコアに、第2問で獲得した  
た ひつよう ふ せいいかい ば あい  
スコアを足す必要がある。不正解の場合は、スコアは0から変化しないから、incorrect  
かんすう かん めいれい け  
関数のスコアに関する命令は消しておこう。

つぎ が そう じっこう かんすう し てい おな  
次に画像をタップしたら実行する関数を指定しよう。ボタンと同じようにonclickを  
つい か が そう じっこう かんすう し てい  
追加すれば、画像をクリックしたときに実行する関数を指定することができるぞ。ピコ  
が そう かんすう じっこう  
の画像 (pico.png) だけは correct 関数を実行するようにして、あの2つは incorrect

かんすう してい  
関数を指定すれば

よい38

```
22 </head>
23 <body>
24 <h1>第2問</h1>
25 <p>Scratchのキャラクター、ピコは? </p>
26 
27 
28 
29 </body>
30 </html>
```

39に第2問のページ(quiz2.html)のHTMLとJavaScriptをまとめておくね。プレビューでも動作を確認して、第3問のページに移動することをチェックしておこう。

```
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,
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        function correct(){
12            alert("すごい！正解だよ！");
13            location.href = "quiz3.html";
14            sessionStorage.score++;
15        }
16
17        function incorrect(){
18            alert("残念！はずれ！");
19            location.href = "quiz3.html";
20        }
21    </script>
22 </head>
23 <body>
24     <h1>第2問</h1>
25     <p>Scratchのキャラクター、ピコは？</p>
26     
27     
28     
29 </body>
30 </html>
```



## 10 3ページ目のHTMLの編集

```

12  </head>
13  <body>
14  | <h1>第3問</h1>
15  | <p>JavaScriptで変数を作るときに使う命令は? </p>
16  | <button onclick="quiz3();">答えを入力</button>
17  </body>
18  </html>

```

だいもん 第3問のページをつくりていこう。まずはHTMLからだね。問題文と、答えを入力するためのボタンを表示するようにしたよ⑩。ボタンを押して実行する関数はquiz3という名前にしたぞ。

40

### 第3問

JavaScriptで変数を作るときに使う命令は?

このボタンを追加



## 11 入力の受け取り

```

9  | <link rel="stylesheet" href="css/style.css">
10 | <script>
11 | function quiz3(){
12 |   var answer = prompt("ヒント：英語で3文字だよ");
13 |   if(answer == "var"){
14 |     alert("すごい！正解だよ！");
15 |   }else{
16 |     alert("残念！はずれ！");
17 |   }
18 | }
19 | </script>
20 | </head>
21 | <body>
22 | | <h1>第3問</h1>
23 | | <p>JavaScriptで変数を作るときに使う命令は? </p>
24 | | <button onclick="quiz3();">答えを入力</button>

```

つぎ 今はquiz3の関数をつくりていこう。プロンプトを表示して答えを入力してもらい、それが正解かどうかを調べるJavaScriptを書いていくぞ⑪。プロンプトを表示して変数に代入するのは、あいさつアプリの応用だ。今回の変数名はanswer(答え)にしたよ。ここで入力された内容がanswerという変数に代入できる。そのanswerの内容を調べて、正解か不正解かを判断する必要があるね。

41

## 条件分岐の書き方

```

if(条件){
  条件が成立するの処理
}else{
  条件が不成立のときの処理
}

```

たいおう [と]を対応させ、この間にじけん せいつ ふせりつ 条件の成立・不成立のそれぞれのはい しょり か場合にしたい処理を書く

42

定義 quiz3

ヒント：英語で3文字だよ と聞いて待つ  
もし 答え = var なら  
すごい！正解だよ！ と言う  
でなければ  
残念！はずれ！ と言う

Scratchの  
ブロックだと…

JavaScriptではifという命令を使って「条件分岐」(41ページ参照)を書くぞ。上に条件分岐の書き方をまとめたので参考にしよう。Scratchのブロックと比べてみたのが⑫だ。Scratchと違うのは、入力された答えを保存しておく変数answerを自分で用意していることと、答えを調べるときの記号が==になっていることだね。

正解のvarという文字列を入力すると、正解と表示されること。違う文字列を入力すると不正解となることを確認しておこう⑬。

43

正解のvarを入力

...007820a206b46a84ad039d--507299523-8080.monaca-app.comに埋め込まれているページの内容  
ヒント：英語で3文字だよ  
 OK キャンセル

...007820a206b46a84ad039d--507299523-8080.monaca-app.comに埋め込まれているページの内容  
ヒント：英語で3文字だよ  
 OK キャンセル

...007820a206b46a84ad039d--507299523-8080.monaca-app.comに埋め込まれているページの内容  
すごい！正解だよ！

...007820a206b46a84ad039d--507299523-8080.monaca-app.comに埋め込まれているページの内容  
残念！はずれ！



## 12 3ページ目の仕上げ

正解の場合も不正解の場合も、最後のページの record.html に移動する。また、正解の場合はスコアを増やすという処理を追加しよう⑭。

⑮に第3問のページ(quiz3.html)のHTMLとJavaScriptをまとめておくね。

```

9  <link rel="stylesheet" href="css/style.css">
10 <script>
11 function quiz3(){
12     var answer = prompt("ヒント：英語で3文字だよ");
13     if(answer == "var"){
14         alert("すごい！正解だよ！");
15         location.href = "record.html";
16         sessionStorage.score++;
17     }else{
18         alert("残念！はずれ！");
19         location.href = "record.html";
20     }
21 }
22 </script>
23 </head>
24 <body>
25 <h1>第3問</h1>
26 <p>JavaScriptで変数を作るときに使う命令は？</p>

```

44

```

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,
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     function quiz3(){
12         var answer = prompt("ヒント：英語で3文字だよ");
13         if(answer == "var"){
14             alert("すごい！正解だよ！");
15             location.href = "record.html";
16             sessionStorage.score++;
17         }else{
18             alert("残念！はずれ！");
19             location.href = "record.html";
20         }
21     }
22 </script>
23 </head>
24 <body>
25     <h1>第3問</h1>
26     <p>JavaScriptで変数を作るときに使う命令は？</p>
27     <button onclick="quiz3();">答えを入力</button>
28 </body>
29 </html>

```

ついで  
追加した  
JavaScript

ついで  
追加した  
HTML

45

## 13 再挑戦ボタンの作成

いよいよ最後の結果表示(record.html)のページだ。まずはもう一度クイズに挑戦できるようにするためのボタンを用意して、そのボタンが押されたら実行される reset という名前の関数をつくろう。reset の関数では、第1問を出題する index.html に戻るようにして、スコアを0に戻しておくよ⑯。

```

9  <link rel="stylesheet" href="css/style.css">
10 <script>
11 function reset(){
12     location.href = "index.html";
13     sessionStorage.score = 0;
14 }
15 </script>
16 </head>
17 <body>
18     <button onclick="reset();">もう一度最初から</button>
19 </body>
20 </html>

```

46

## 14 コメント表示

次はスコアに応じたコメントを表示する部分をつくろう。関数として

つくった処理は、ページが読み込まれたときには実行されず、関数を呼び出すことが必要だったね。script のタグに書かれた JavaScript の命令は関数としてつくった以外の部分の命令は、ページが読み込まれたときに実行される。最初に結果発表という文字を画面に表示する命令をつけ加えて、動作を確認してみよう⑰。追加するのは、document.

write という命令だよ。これは画面に指定した文字列を表示するものだ。この命令は reset 関数の上に書いてあるぞ。script タグの中に書かれた命令は基本的に上から順番に実行されていくけれど、関数としてつくった部分は呼び出しがなければ実行はされない

```

9  <link rel="stylesheet" href="css/style.css">
10 <script>
11     document.write("<h1>結果発表</h1>");
12
13 function reset(){
14     location.href = "index.html";
15     sessionStorage.score = 0;
16 }
17 </script>

```

47

ページが読み込まれたら実行

```

9  <link rel="stylesheet" href="css/style.css">
10 <script>
11     document.write("<h1>結果発表</h1>");
12
13 function reset(){
14     location.href = "index.html";
15     sessionStorage.score = 0;
16 }
17 </script>
18 </head>
19 <body>
20     <button onclick="reset();">もう一度最初から</button>

```

reset 関数の呼び出しがあるまで実行されない

48

```

9  <link rel="stylesheet" href="css/style.css">
10 <script>
11 document.write("<h1>結果発表</h1>");
12
13 function reset(){
14   location.href = "index.html";
15   sessionStorage.score = 0;
16 }
17 </script>
18 </head>
19 <body>
20 <button onclick="reset();">もう一度最初から</button>

```

script タグの後の HTML も上から順番に表示されていくから、今回は結果発表という文字は JavaScript で表示させて、ボタンは HTML で表示させているということになるね④。

なぜこんなことをしているかというと、結果発表という表示は毎回共通だけれど、スコアは sessionStorage.score に保存してあり、コメントはその値によって変化をさせ必要があるからだ。HTML にスコアに関するコメントを書いてしまっては、毎回同じ表示になってしまふ。だから JavaScript からコメントを表示させる必要があるんだね。

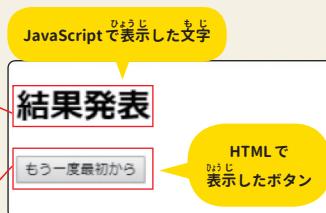
```

9  <link rel="stylesheet" href="css/style.css">
10 <script>
11 document.write("<h1>結果発表</h1>");
12 document.write("<h2>正解は " + sessionStorage.score + " 問でした</h2>");
13 if(sessionStorage.score == 3){
14   document.write("<p>全問正解！大変よくできました！</p>");
15 }else{
16   document.write("<p>もう一度挑戦して、全問正解を目指そう！</p>");
17 }
18
19 function reset(){
20   location.href = "index.html";
21   sessionStorage.score = 0;
22 }
23 </script>
24 </head>
25 <body>
26 <button onclick="reset();">もう一度最初から</button>

```

ではスコアとそれに応じたコメントを表示する部分をつけ加えよう。今回はシンプルに3問正解だった場合とそれ以外の場合に分けてみたよ⑤。record.html については、HTML と JavaScript のまとめはないけれど、⑤をよく見て確認しよう。クイズアプリでつくる HTML と JavaScript は、8ページで紹介しているサポートページからダウンロードもできるよ。うまく動作しないときはサポートページを見てね。

49



## 15 動作結果をチェック

これでクイズアプリは完成だ。Monaca デバッガーを使って、全問正解の場合⑤、そうでない場合の両方の動作確認をしておこう。

51

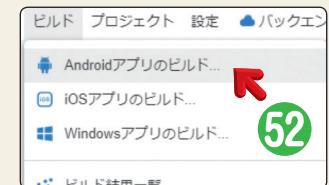


## 16 アプリのビルトの準備

Monaca デバッガーでもアプリの動作を確認することができるけれど、Android のスマホを使っている場合、ほかのアプリと同じように、クイズアプリとして動作させることができるぞ。必要なファイルを準備して、アプリの準備をすることを「ビルト」というよ。

まずはビルトのメニューから「Android アプリのビルト」を選択しよう⑥。

アプリの設定の Android の項目を選択して⑦、アプリケーション名とパッケージ名を変更しよう。パッケージ名は元から入力されている「com.example.helloworld」を少し変更するだけで大丈夫だよ⑧。



52



54



## 17 アイコンの設定

次にアプリのアイコンを設定しよう。アイコンにしたい画像をpng形式で用意しよう。ある程度の大きさの画像を用意して、一括で設定するの下にある「アップロード」を選択し<sup>55</sup>、アイコンとして設定したい画像ファイルを選択しよう。

クイズアプリなので、はてなマークの画像をアップしているよ。アイコンの設定が終わったら「保存する」をクリックしよう<sup>56</sup>。



55



56



## 18 ビルドの開始とインストール

画面の左から、ビルド/ビル  
ド設定のAndroidを選択し  
う。デバッグビルドを選択して  
57、「ビルドを開始する」のボ  
タンを押すよ<sup>58</sup>。新しい画面が開  
いて処理がはじまるので、しば  
らく待とう。

ビルドがうまくいくと、QR  
コードなどが表示されたページ  
に切り替わるぞ。一番手軽な  
のは、画面に表示されたQRコードをスマホで読み取ってアプリをインストールする方法だ<sup>59</sup>。



57

58



59

QRコードを読み取れば、アプリのファイルがダウンロードできるよ<sup>\*</sup>。ダウンロードしたファイルを選択して<sup>60</sup>、インストールをタップしよう<sup>61</sup>。ダウンロードの際に、Monacaのサイトにサインインするように指示された場合は、メールアドレスとパスワードを入力しよう。「アプリをインストールしました。」の表示がでれば、アプリのインストールは完了だね<sup>62</sup>。



55

56

58

59

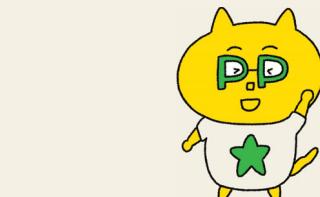
ほかのアプリと同じようにクイズアプリが使えるようになるぞ<sup>63</sup>。なお、今使っている無料のプランだと、1日に3回までしかビルドができないから注意しよう。

iPhoneなどのiOSが動作しているスマートフォンについては、ビルドにアップル社から発行してもらう証明書が必要なので、少し準備が大変だよ。詳しい方法については以下のMonacaのマニュアルを見てみよう。

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



63



スマホアプリをつくるのはどうだったかな？  
シンプルなクイズアプリをつくるってみたけれど、  
いろいろ改造して、オリジナルの  
クイズアプリにしてみよう！

\*アプリがインストールできない場合は、「設定」のメニューから「ユーザ設定」の「セキュリティ」を選択し、「提供元不明のアプリ」を一時的に許可しよう。インストールが完了したら「提供元不明のアプリ」の設定は元に戻すようにすること。