

「情報 I」に向けたプログラミング研修会

～文科省教員研修用教材(JavaScript版)の
ポイントを踏まえたプログラミング入門～

アシアル株式会社
アシアル情報教育研究所



自己紹介

■名前

- 井澤哲也(アシアル情報教育研究所)

■メッセージ

- Monacaによるアプリ開発を通じて、情報技術の活用方法や作品作りの楽しさを広めてまいります。



アシアル株式会社について

- **2002年**

- 代表の田中正裕が本郷の地で創業(当時20才)
- PHP言語に関する雑誌発刊・教育・コンサルティング

- **2010年**

- アシアルPHPスクールのマネージャーに岡本が就任

- **2012年**

- アプリ開発ツール「Monaca」をリリース
- JavaScript言語とHTML5による複数OS向けアプリ開発環境を提供

- **2015年**

- MonacaEducation事業がスタート

- **2020年**

- アシアル情報教育研究所設立

Monaca Education

<https://edu.monaca.io>

アプリ開発によるプログラミング教育

ログイン

アカウント作成

 Monaca Education

Monacaで学ぶ理由

教材

教員研修

料金表

FAQ

学習者向け

お知らせ



先生の想いと生徒の未来に寄り添う
プログラミング教育

お問合わせ

Monaca Educationとは

Monaca Educationは、教育機関における情報教育を通じて中高生の未来を拓くプログラミング教育サービスです。生徒たちはスマホアプリを作りながらプログラミングの基礎学習から作品作りまで取り組めます。現場の先生方からは、教材の内容に加えて、研修や技術サポート、補助教材が豊富で安心して授業に取り組めるとご評価いただいています。

あんこエデュケーション

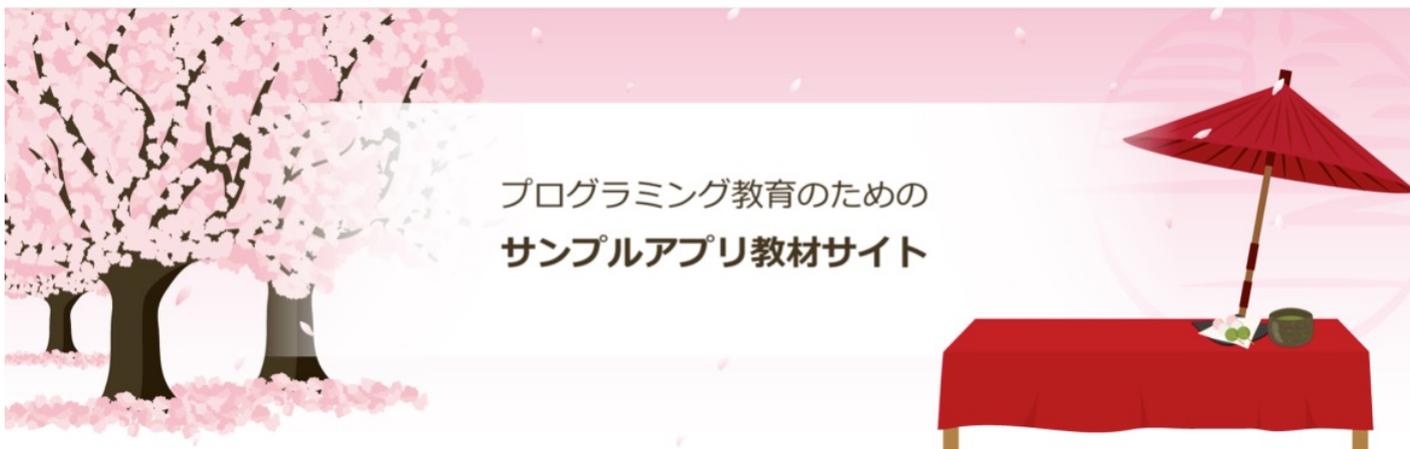
<https://anko.education>

プログラミング教育のためのサンプルアプリ教材サイト

お知らせ アシアル情報教育研究所ブログ



サンプルアプリ集 素材集 ツール集 リファレンス 情報 | 研修資料 学習指導案



Monacaから飛び出した「あんこ」

製品の枠を越えてプログラミングや情報教育に役立つ情報をお届けします。



激甘モード

初心者を甘やかすことを最優先します。激甘リファレンスも準備中。



有料広告はありません

国産クラウド型アプリ・プログラミング教材 MonacaEducationの自社媒体として運営しています。

今日の内容

- 次期学習指導要領におけるプログラミングの位置付け
- 文科省教員研修用教材・第3章
「コンピュータとプログラミング」ポイント解説
- Webで学ぶ論理回路 (※SimcirJSを使用)
- Webで学ぶ外部装置 (※Microsoft MakeCode/micro:bitを使用)
- JavaScriptで桁あふれ
- JavaScriptで複利計算
- JavaScriptでグラフ表示
- JavaScriptでWebAPI
- Monaca利用事例の紹介

- JavaScriptとは
- Monacaで出来ること

次期学習指導要領における プログラミングの位置付け

次期学習指導要領とプログラミング

- 小学校(2020～)

- 既存の教科の中で実施

- 中学校(2021～)

- 技術・家庭科の中で実施

- 高校(2022～)

- 教科「情報」で実施
- 共通必修科目「情報Ⅰ」が2単位(70時間)

中学校のプログラミング教育

- 以前
 - プログラムによる計測・制御
- 2021年度から
 - プログラムによる計測・制御
 - 「ネットワークを利用した双方向性のあるコンテンツのプログラミング」

高校のプログラミング教育

• 現行

- 教科「情報」は2003年度から存在する
- 現行科目は「社会と情報」と「情報の科学」の選択必修
- プログラミングを扱うのは「情報の科学」
- 「情報の科学」は2割程度の学校が履修

• 次期

- 新科目「情報Ⅰ」が必修に。学習範囲にはプログラミングが含まれる
- 更に新科目「情報Ⅱ」が選択科目として追加される

※「総合的な探究の時間」でSTEM教育が取り入れられる。

他の科目でもプログラミングが扱われる可能性がある

情報 I

情報に関する**科学的な見方・考え方**を働かせ、情報技術を活用して問題の発見・解決を行う学習活動を通して、**問題の発見・解決**に向けて情報と情報技術を適切かつ**効果的に活用**し、情報社会に**主体的に参画**するための資質・能力を次のとおり育成することを目指す。

- (1) 情報社会の問題解決
- (2) コミュニケーションと情報デザイン
- (3) コンピュータとプログラミング
- (4) 情報通信ネットワークとデータの活用

文科省教員研修用教材・第3章
「コンピュータとプログラミング」ポイント解説



文科省教員研修用教材

高等学校情報科教員研修用教材 - 高等学校情報科 (各学科に共通する教科)

https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416746.htm

- (1) 情報社会の問題解決
- (2) コミュニケーションと情報デザイン
- (3) コンピュータとプログラミング
- (4) 情報通信ネットワークとデータの活用

第三章の内容と、この資料で取り上げる内容

学習	タイトル	ポイント	この後のセクション
11	コンピュータの仕組み	論理回路 桁あふれ	Webで学ぶ論理回路
12	外部装置との接続	micro:bitなど 順次・分岐・繰り返し	Webで学ぶ外部装置
13	基本的プログラム	変数・順次・分岐・繰り返し	JavaScriptとは
14	応用的プログラム	関数・乱数・配列・WebAPI	Monacaでできること
15	アルゴリズムの比較	複数の分岐と繰り返し	Monacaで簡単なアプリを動かす
16	確定モデルと確率モデル	多数のデータの取り扱い・グラフ	JavaScriptで桁あふれ
17	自然現象のモデル化とシミュレーション	多数のデータの取り扱い・グラフ	JavaScriptで複利計算 JavaScriptでグラフ表示 JavaScriptでWebAPI

Webで学ぶ論理回路

>> Next

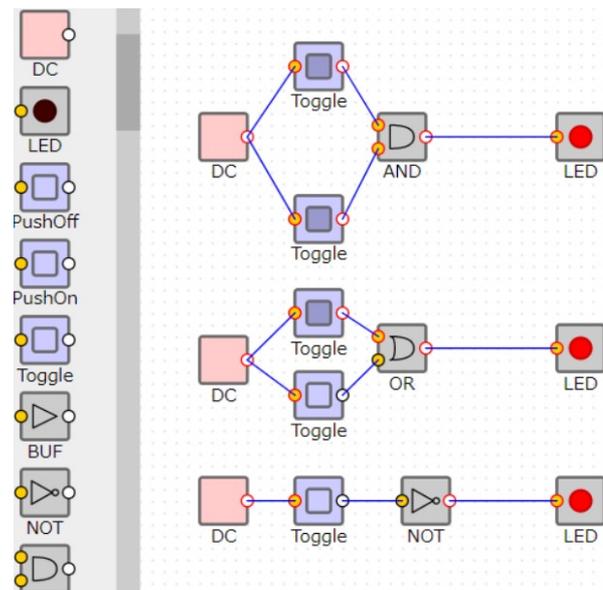
- Webで学ぶ外部装置
- JavaScriptとは
- Monacaでできること
- JavaScriptで桁あふれ
- JavaScriptで複利計算
- JavaScriptでグラフ表示
- JavaScriptでWebAPI



【実習】AND回路とOR回路とNOT回路体験

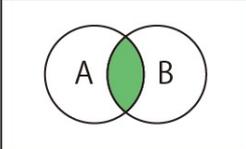
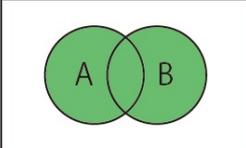
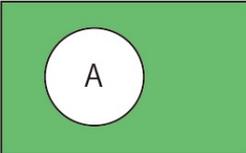
• 論理回路シミュレータ SimcirJS

- ブラウザ上で論理回路の組み立てを行える
- ライセンスはオープンソース(MIT)
- 本実習で必要な装置
 - DC(入力)
 - Toggle (on/off)
 - AND
 - OR
 - NOT



【実習】AND回路とOR回路とNOT回路体験

- 教員研修資料の挙動を確認してみよう

A and B AとBの両方を含む		A and B (A,Bはそれぞれ0または1) 0 and 0 = 0 (なしとなしの重なりはなし) 0 and 1 = 0 (なしとありの重なりはなし) 1 and 0 = 0 (ありとなしの重なりはなし) 1 and 1 = 1 (ありとありの重なりはあり) ※ and はかけ算に似ている。and を論理積という。	<table border="1"><thead><tr><th>A</th><th>B</th><th></th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	A	B		0	0	0	0	1	0	1	0	0	1	1	1
A	B																	
0	0	0																
0	1	0																
1	0	0																
1	1	1																
A or B AとBのどちらかを含む		A or B (A,Bはそれぞれ0または1) 0 or 0 = 0 (なしとなしのどちらかはなし) 0 or 1 = 1 (なしとありのどちらかはあり) 1 or 0 = 1 (ありとなしのどちらかはあり) 1 or 1 = 1 (ありとありのどちらかはあり) ※ or は足し算に似ている。or を論理和という。	<table border="1"><thead><tr><th>A</th><th>B</th><th></th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	A	B		0	0	0	0	1	1	1	0	1	1	1	1
A	B																	
0	0	0																
0	1	1																
1	0	1																
1	1	1																
not A Aではない		not A (Aは0または1) not 0 = 1 (なしではないものはあり) not 1 = 0 (ありではないものはなし) ※ not で0は1,1は0になる。not を否定という。	<table border="1"><thead><tr><th>A</th><th></th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	A		0	1	1	0									
A																		
0	1																	
1	0																	

図表 4 論理演算

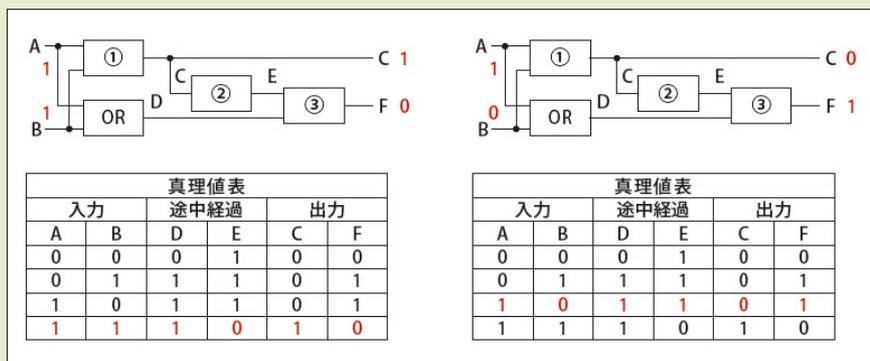
※高等学校情報科「情報 I」教員研修用教材 JavaScript版 p5

【実習】AND回路とOR回路とNOT回路体験

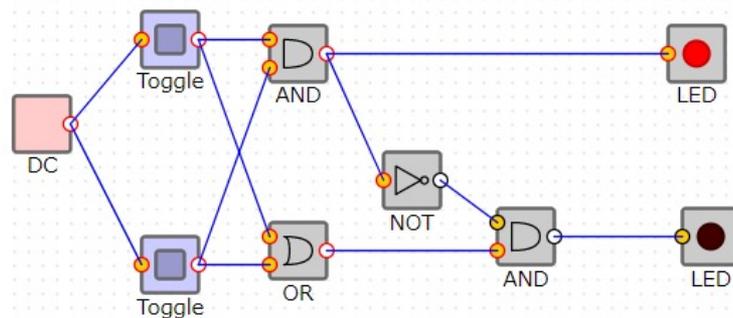
- 演習2で示されている「二進数の足し算」に挑戦

<演習2>

次の論理回路は入力AとBがともに1の場合、出力Cが1、Fが0となります。また入力Aが1、Bが0の場合、出力Cが0、Fが1となるものです。この論理回路の①、②、③に適切な論理演算を入れ、上記の二進数の足し算を成立させましょう。



図表5 論理回路と真理値



※高等学校情報科「情報I」教員研修用教材 JavaScript版 p6

Webで学ぶ外部装置

>> Next

- JavaScriptとは
- Monacaでできること
- JavaScriptで桁あふれ
- JavaScriptで複利計算
- JavaScriptでグラフ表示
- JavaScriptでWebAPI



【実習】外部装置

- **Microsoft MakeCode**

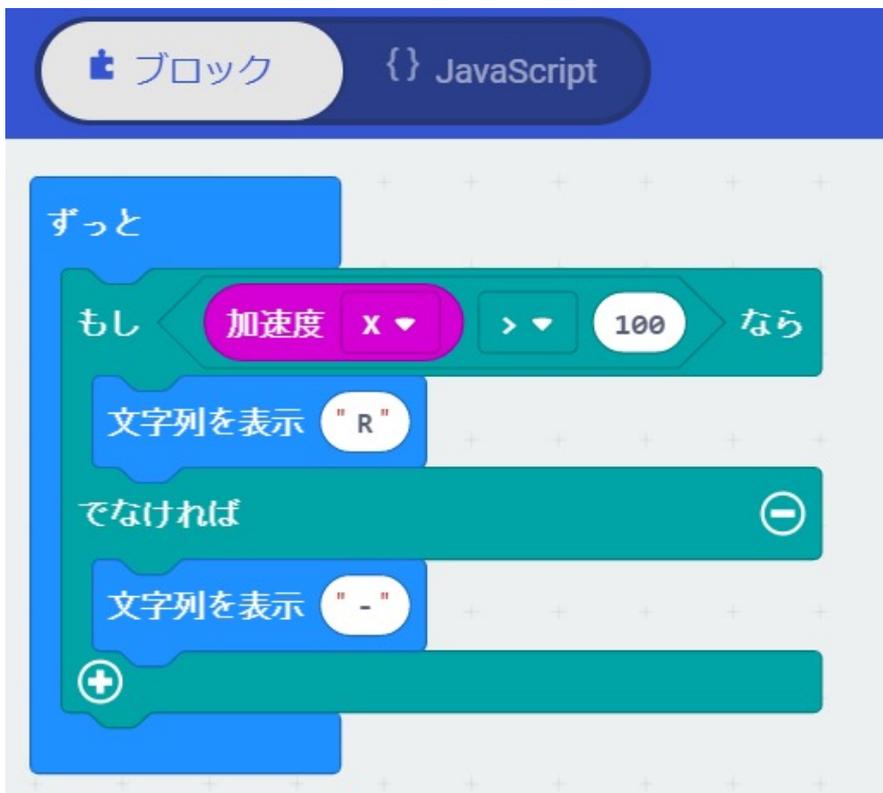
<https://makecode.microbit.org/>

- ブラウザ上でMicro:bitのプログラミングができる
- エミュレータも付いている
- JavaScriptとブロックの相互変換も可能



【実習】外部装置で分岐と反復

分岐



Scratch code for branching logic. The code is in JavaScript mode. It starts with a 'ずっと' (Forever) loop block. Inside the loop, there is an 'もし' (If) block with the condition '加速度 x > 100 なら' (If acceleration x > 100). Inside the 'もし' block, there are two '文字列を表示' (Say) blocks: one with the text 'R' and another with the text '-'. The 'もし' block has a '+' sign on its right side, indicating it is an 'if-then-else' structure.

繰り返し



Scratch code for loop logic. The code is in JavaScript mode. It starts with an '最初だけ' (Once) block. Inside, there is a 'くりかえし 10 回' (Repeat 10 times) loop block. Inside the loop, there are four blocks: '変数 i を 1 だけ増やす' (Increase variable i by 1), '文字列を表示 i' (Say i), '一時停止 (ミリ秒) 1000' (Wait 1000 ms), and '表示を消す' (Clear display). The loop block has a '+' sign on its right side, indicating it is a 'repeat with' loop.

【参考】分岐と反復のJavaScriptソースコード

分岐

```
basic.forever(function () {  
  if (input.acceleration(Dimension.X) > 100) {  
    basic.showString("R")  
  } else {  
    basic.showString("-")  
  }  
})
```

繰り返し

```
let i = 0  
for (let i0 = 0; i0 < 10; i0++) {  
  i += 1  
  basic.showString("" + i)  
  basic.pause(1000)  
  basic.clearScreen()  
  basic.pause(1000)  
}
```

※高等学校情報科「情報 I」教員研修用教材 JavaScript版 p.6-7

if文とfor文では【条件式】を利用している

繰り返し(while文)の紹介

```
1 let i = 0
2 while (true) {
3     i += 1
4     basic.showString("" + i)
5     basic.pause(1000)
6     basic.clearScreen()
7     basic.pause(1000)
8 }
9
```

for文から条件式以外を取り除けばwhile文に!

JavaScriptとは

>> Next

- Monacaでできること
- JavaScriptで桁あふれ
- JavaScriptで複利計算
- JavaScriptでグラフ表示
- JavaScriptでWebAPI



JavaScriptの特徴(1)

- **ブラウザ上でプログラムを動作させる**

- HTML言語だけではブラウザ上のコンテンツ(画像や表など)を静的に表示することしかできない
- JavaScriptがあれば画面に動きをつけることができる
- HTML(構造のある文書)+CSS(要素の表示方法)+JavaScript(動作)

- **外部のコンピュータと通信できる**

- 外部のサーバと通信できる。高度なアプリケーションを開発できる
 - 代表例:Google Maps
 - 地図の画像データはクラウド側にある
 - 手元のコンピュータのブラウザは操作性と、受信した画像データの表示を行う

※もはやブラウザさえあればOS不要では? ⇨ Chrome OS

JavaScriptの特徴(2)

• クロスプラットフォーム

- ブラウザさえあればどのOSでも利用できる
- 標準化団体Ecma Internationalが標準を定義している
- ブラウザの開発者は、この標準を参照して、JavaScriptの実行部分を作っている
- 後方互換性も維持されている

※ブラウザごとの動作の差異が残っていることには注意が必要

ブラウザ上で動作するメリット

- **利用者**

- インストール無しで気軽にサービスを利用できる

- **アプリ開発者**

- OSのストアを経由せずにアプリを配信できる
- 一つの言語で複数OSにアプリを出せる

参考：幾つかの言語と成り立ち

- **BASIC言語**

- Beginner向けの言語として1964年に登場

- **C言語**

- 主にOS (UNIX) 開発のため1972年頃に登場

- **VBA**

- MS Office上で動作する言語として1993年に登場

- **JavaScript**

- ブラウザ上で動くプログラミング言語として1995年に登場

言語も、何らかの課題を解決するために登場しています

教育現場におけるJavaScriptのメリット

- **生徒の反応が良い**
 - 画像や音を簡単に扱える
- **学校の外でも気軽に取り組める**
 - 生徒が自宅で予習や復習しやすい
 - 研修もブラウザベースの方が組みやすい
- **教材の寿命が長い**
 - 指導案やプリントを使い続けられる

Monacaでできること

>> Next

- JavaScriptで桁あふれ
- JavaScriptで複利計算
- JavaScriptでグラフ表示
- JavaScriptでWebAPI



Monacaの特徴

ブラウザだけで動作するクラウドIDE(統合開発環境)

The image displays the Monaca IDE interface, which is a cloud-based development environment. The interface is divided into several sections:

- メニューバー (Menu Bar):** Located at the top, it contains various menu items such as 'ファイル' (File), '編集' (Edit), '表示' (View), '実行' (Run), 'ビルド' (Build), 'プロジェクト' (Project), '設定' (Settings), 'バックエンド' (Backend), and 'ヘルプ' (Help).
- プロジェクトパネル (Project Panel):** Located on the left side, it shows a file tree structure for the current project, including folders like 'node_modules', 'rs', 'www', and 'assets', and files like 'questions.csv', 'app.js', 'index.html', and 'quiz.html'.
- コードエディター (Code Editor):** The central area where the source code is written. It shows HTML and JavaScript code for a mobile application. A callout box indicates that this is the 'コードエディター (ファイルの編集を行います)' (Code Editor (File editing)).
- デバッグパネル (Debugger Panel):** Located at the bottom, it provides tools for debugging, including 'Elements', 'Resources', 'Network', 'Timeline', and 'Console'. A callout box indicates that this is the 'デバッグパネル (エラー情報などを表示します)' (Debugger Panel (Displays error information, etc.)).
- ライブプレビュー (Live Preview):** On the right side, there is a preview window showing the rendered mobile application on an iPhone 8. The app has a blue background with a white question mark icon and a 'スタート' (Start) button. A callout box indicates that this is the 'ライブプレビュー (動作確認を行います)' (Live Preview (Action confirmation)).

Additional callouts include:

- 'メニューバー (さまざまな機能呼び出せます)' (Menu Bar (Calls out various functions))
- 'プロジェクトパネル (ファイルの管理などを行います)' (Project Panel (Manages files, etc.))
- 'デバッグパネル (エラー情報などを表示します)' (Debugger Panel (Displays error information, etc.))
- 'ライブプレビュー (動作確認を行います)' (Live Preview (Action confirmation))

Monacaの特徴

Monaca for Study (アプリ) でスマホやタブレットが教材に



MonacaIDE上で記述したソースコードが手元で動き出す!
QRコードの読み込みも可能(2020年度より)

Monacaの特徴

外部のWebAPIを利用可能



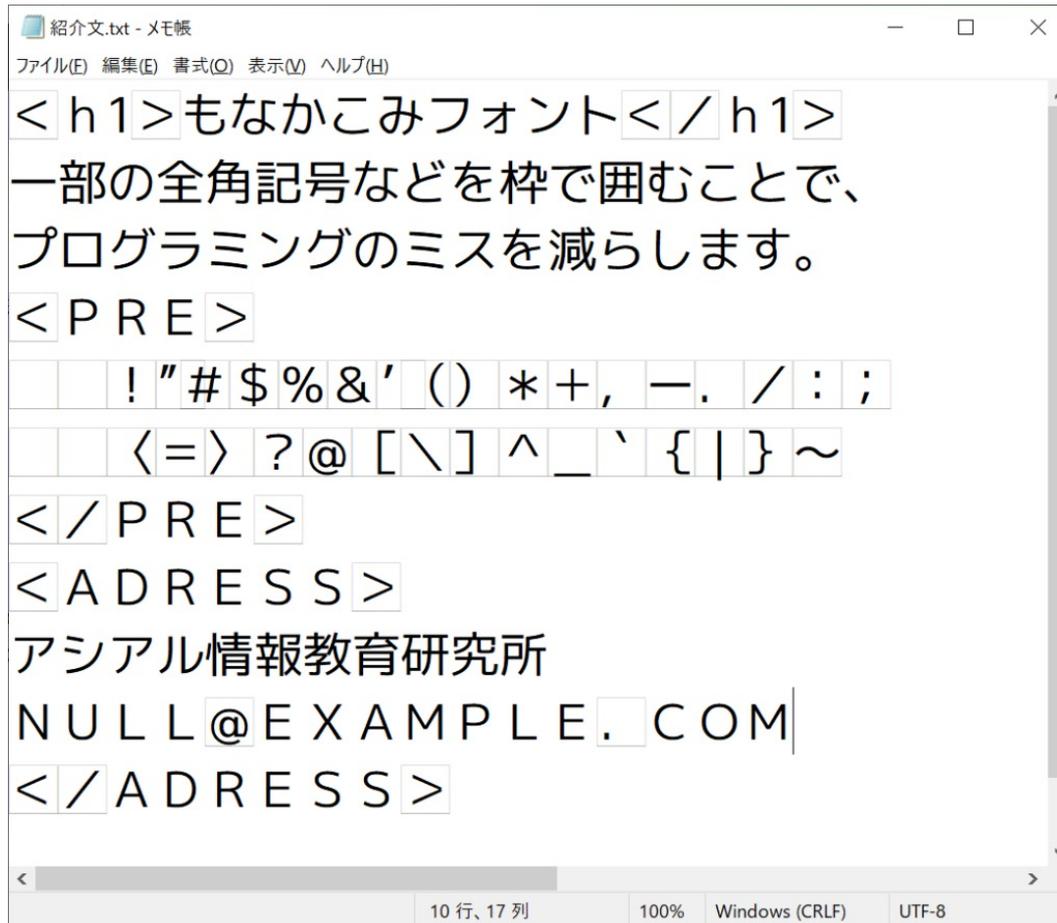
JavaScriptによる
データの送受信



簡易データベース機能
(※Monaca Educationのみ)

外部のWebAPI

もなかこみフォント搭載(2020年度より)



```
紹介文.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
< h 1 > もなかこみフォント < / h 1 >
一部の全角記号などを枠で囲むことで、
プログラミングのミスを減らします。
< P R E >
! " # $ % & ' ( ) * + , - . / : ;
< = > ? @ [ \ ] ^ _ ` { | } ~
< / P R E >
< A D R E S S >
アシアル情報教育研究所
NULL @ E X A M P L E . C O M
< / A D R E S S >
10行、17列 100% Windows (CRLF) UTF-8
```



Monacaで簡単なアプリを動かす

>> Next

- JavaScriptで桁あふれ
- JavaScriptで複利計算
- JavaScriptでグラフ表示
- JavaScriptでWebAPI



ダッシュボード

- 開発中のアプリはプロジェクト単位で管理します
- 新しいプロジェクトを作ると一覧に並びます



新しいプロジェクトを作る

- 今回は「ブロック崩し」を選択して作ってみましょう
- プロジェクト名は自由に設定できます

⊗ 新しいプロジェクトを作る

1 テンプレート

 <p>HelloWorld!</p> <p>Start Demo</p> <p>HelloWorld</p> <p>HelloWorldを表示するだけのアプリです。</p>	 <p>SCORE: 0</p> <p>ブロック崩し</p> <p>pixi.jsを用いたブロック崩しゲームです。難易度やシナリオはプレイ後に自由に調整してください。</p>	 <p>This is a template for Monaco app.</p> <p>最小限のテンプレート</p> <p>フレームワークを使用しない空のテンプレートです。</p>
--	--	---

2 プロジェクトの情報

⊗ 新しいプロジェクトを作る

テンプレート ブロック崩し

2 プロジェクトの情報

プロジェクト名
ブロック崩し

説明

作成

プロジェクトを開く

- 一覧からプロジェクトを選択して下さい
- 「クラウドIDEで開く」を選択して下さい

Monaca Education

アカウント管理 Monacaを学ぶ お問い合わせ

新しいプロジェクトを作る インポート

タグ オンライン 作成日時

ブロック崩し
No project description.

NEW

ブロック崩し

プロジェクトの概要がありません
作成日時: 数秒前 / 最終アクセス: 数秒前 / フレームワーク: Cordova 7.1.0

<> 開発 🛠️ ビルド ⚙️ 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

TIPS

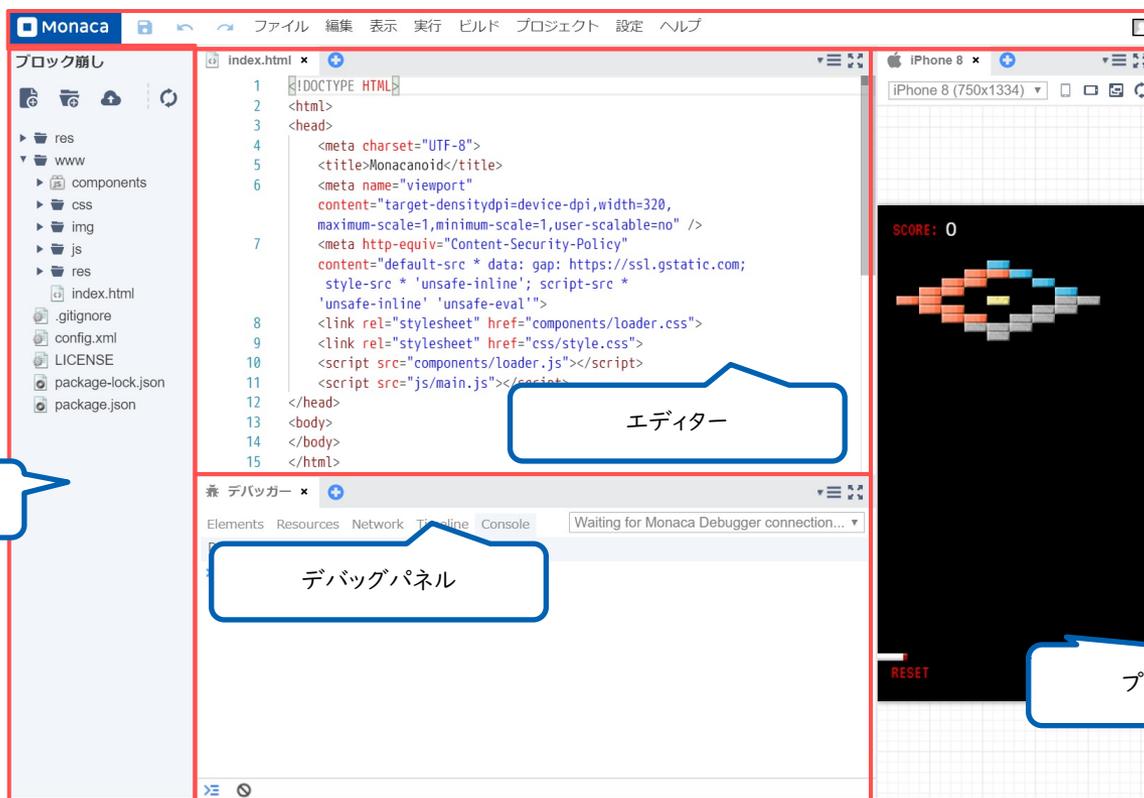
無限ループなどで開けなくなったプロジェクトはセーフモードで開けます。

クラウドIDEで開く

セーフモードで開く

MonacaIDE(統合開発環境)

- ファイル管理・編集・動作確認などを行えます



フォルダとファイルを開く

プロジェクトパネルのファイルツリーを操作して下さい

ブロック崩し

res
www
 components
 css
 img
 js
 res
index.html
.gitignore
config.xml
LICENSE
package-lock.json
package.json

js
main.js

ブロック崩し

index.html main.js x +

```
1 (function() {  
2  
3 var SETTINGS_GRAVITY = 0.07,  
4     SETTINGS_FPS = 30,  
5     SETTINGS_BALL_NUM = 1,  
6     SETTINGS_BOUND_X = 0.13,  
7     SETTINGS_BOUND_Y = 1.04,  
8     SETTINGS_ACCELEROMETER_RELOAD_FREQ = 100,  
9     SETTINGS_PADDLE_ACCEL = 2.8,  
10    SETTINGS_POINT = 1000,  
11    SETTINGS_POINT_SILVER = 200,  
12    SETTINGS_POINT_GOLD = 3000000;  
13  
14 var GAMESTATE_STOP = 0,  
15     GAMESTATE_PLAY = 1;
```

フォルダの▶をクリック

ファイルを
ダブルクリック

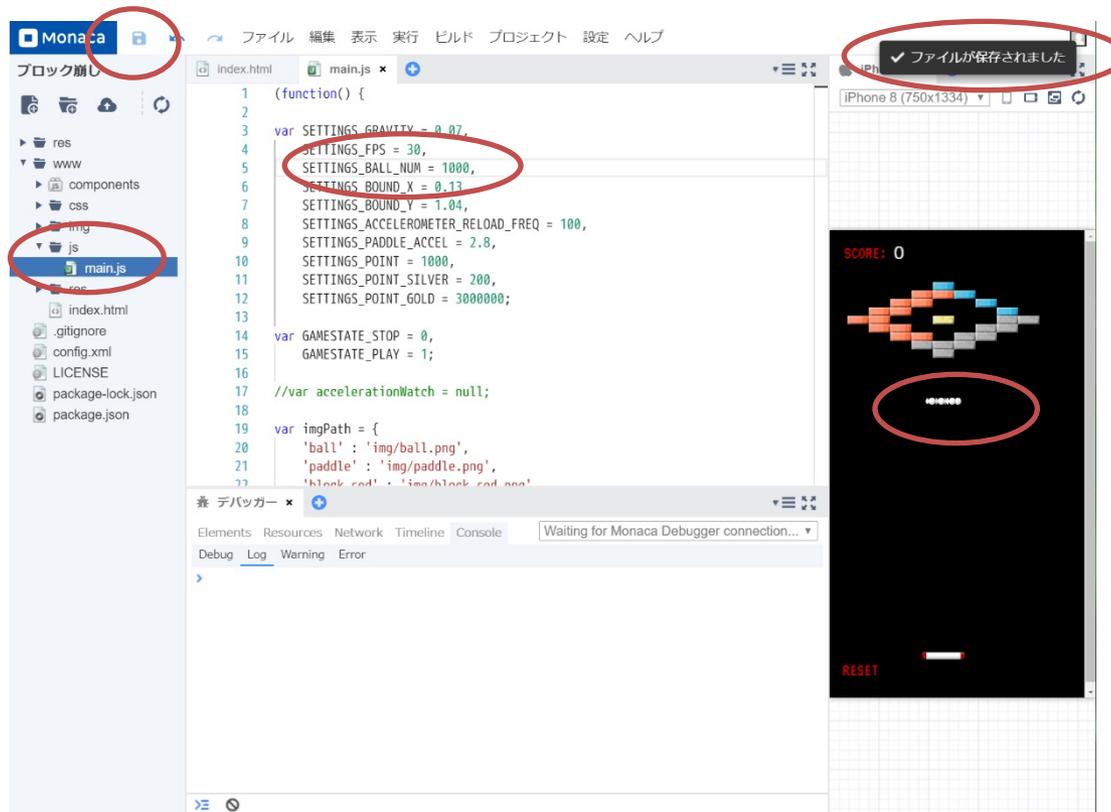
エディタ側の
タブが開きます

プログラムの変更（玉の数を増やす）

- ブロック崩しプログラムの本体は「main.js」ファイルです
- プロジェクトパネルのファイルツリーからmain.jsを探して「ダブルクリック」して下さい
- 5行目の変数SETTINGS_BALL_NUMを1000に「修正」して下さい
- main.jsの変更を「保存」して下さい。
- プレビュー上の玉の数が増えれば成功です

玉が増えれば成功

「保存」はボタンかショートカットキー (Ctrl + s)



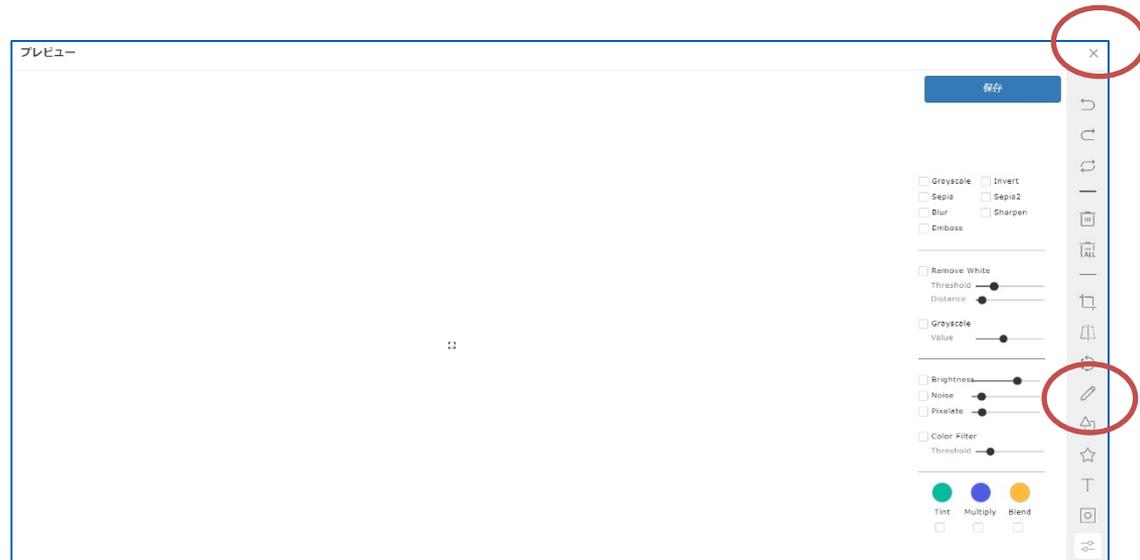
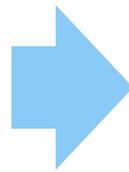
応用：ブロックの配置を変える

- ブロックは二次元配列で管理されています
- Monaca入門のタイミングで配列の予習が可能
- 'red'や'gold'は文字なので「'」か「'''」で囲む

```
index.html main.js x +
43 setMap: function() {
44     var blockMap = [
45         [null, null, null, null, 'blue', null, null, null, null],
46         [null, null, null, null, 'red', 'red', 'blue', null, null],
47         [null, null, null, 'red', 'red', null, null, 'blue', null],
48         [null, null, 'red', 'red', null, null, null, 'blue', null],
49         [null, 'red', 'red', null, null, null, 'gold', null, 'silver', 'silver'],
50         [null, null, 'red', 'red', null, null, null, 'silver', 'silver', null],
51         [null, null, null, 'red', 'red', null, 'silver', 'silver', null, null],
52         [null, null, null, null, 'silver', 'silver', 'silver', null, null, null],
53         [null, null, null, null, null, 'silver', null, null, null, null]
54     ];
55
56     for(j = 0; j < blockMap.length; j++) {
57         for(i = 0; i < blockMap[j].length; i++) {
58             if(blockMap[j][i] !== null) {
59                 var block = BB.addBlock(10 + (30 * i), 80 + (12 * j), blockMap[j][i]);
60             }
61         }
62     }
63 }
```

応用：玉の色を変える(簡易編集)

- 画像ファイルをダブルクリックすると編集できます
- ペンで塗りつぶして下さい
- 保存後に閉じない場合は手動で×をクリック



JavaScriptで桁あふれ

>> Next

- JavaScriptで複利計算
- JavaScriptでグラフ表示
- JavaScriptでWebAPI



【実習】桁あふれ

```
<script>
  x = 1.7976931348623157e+308;
  document.writeln(x+"<br>");
  x = 1.7976931348623157999999e+308;
  document.writeln(x+"<br>");
  x = 1.8e+308;
  document.writeln(x+"<br>");
</script>
```

```
<script>
  x = 28-27;
  document.write(x+"<br>");
  y = 0.28-0.27;
  document.write(y+"<br>")
</script>
```

※高等学校情報科「情報 I」教員研修用教材 JavaScript版 p.7-8

JavaScriptで複利計算

>> Next

- JavaScriptでグラフ表示
- JavaScriptでWebAPI



【実習】複利計算

```
<script>
  yokin = 100000;
  riritsu = 0.05;
  for(i = 0;i < 10;i++) {
    risoku = yokin*riritsu;
    yokin = yokin+risoku
    document.write(i+1," 年目:",yokin,"<br>");
  }
</script>
```

※教員研修用教材 P43より

JavaScriptでグラフ表示

>> Next

- JavaScriptでWebAPI



グラフを作成するために必要な知識

- グラフ化したい値を「配列」に格納する
- 配列を「グラフライブラリ」の関数などに渡す
 - 今回は、Plotlyというライブラリを利用
<https://plotly.com/javascript/>
- 「任意の場所」に描画させる

グラフ描画例

HTML側

グラフライブラリ読み込み

```
<script src="https://cdn.plot.ly/plotly-latest.min.js">
</script>
<body onload="plot()">
  <div id="myDiv"></div>
</body>
</html>
```

HTMLをロードしたら
plot()関数を呼び出す

後でココにグラフを描画

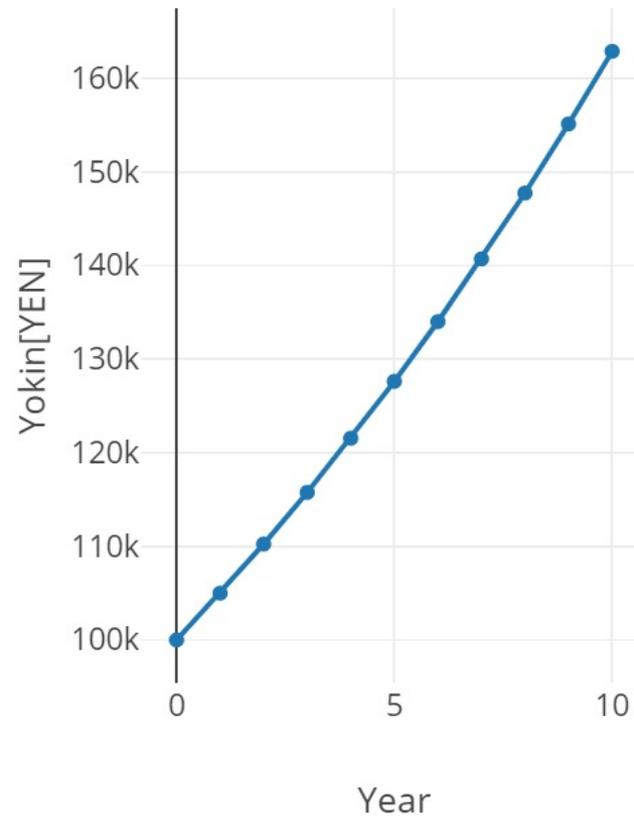
配列のイメージとグラフの結果

配列のイメージ

[9]	162889
[8]	155132
[7]	147745
[6]	140710
[5]	134009
[4]	127628
[3]	121550
[2]	115762
[1]	110250
[0]	100000

グラフの結果

FUKURI KEISAN



グラフ描画例

JS側

```
<script>
function plot() {
  var riritsu = 0.05;
  var yokin = [100000];
  for (var i = 0; i < 10; i++) {
    var risoku = (yokin[i] * riritsu);
    yokin.push(yokin[i] + risoku);
  }
  var trace1 = {
    y: yokin,
    mode: 'lines+markers',
    type: 'scatter'
  };
  var layout = {
    title: "FUKURI KEISAN",
    xaxis: {title: "Year"},
    yaxis: {title: "Yokin[YEN]"}
  };
  var data = [trace1];
  Plotly.newPlot('myDiv', data, layout);
}
</script>
```

① 毎年10万円を複利で預金
※年ごとに配列を追加して値を記録

② 変数trace1にグラフのY軸の値や形式を設定

③ 変数layoutにグラフ全体とX軸・Y軸のタイトルを設定

④ グラフの設定と値を元にプロットを実行
※HTML側のidがmyDivの箇所に描画

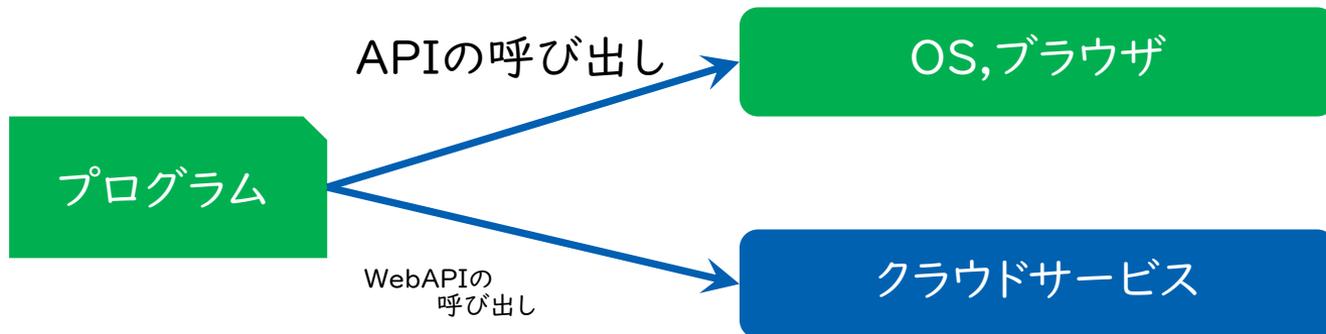
JavaScriptでWebAPI

>> Next
Monaca利用事例の紹介



WebAPIとは

- API (Application Programming Interface) ≡ 関数
- OSやブラウザなどの機能をプログラミング言語で呼び出すときには、APIを経由して利用しています
- 例: ファイルの保存や位置情報の取得など



- クラウドサービスなどをWeb経由で利用できるAPIをWebAPIと呼びます
- JavaScriptでは「fetch」命令で簡単に呼び出せます

WebAPIの例

- 郵便番号を元に住所を取得する
- 郵便番号を元に天気予報の取得する
- 画像を元に写っている物の名前を判定する

郵便番号APIの仕様

リクエスト例

タイプ	値
URL	https://api.anko.education/zipcode/?zipcode=1130033 
メソッド	GET

レスポンス

タイプ: JSON

キー	値
code	integer 郵便番号
prefcode	integer 都道府県コード
pref	String 都道府県名
city	String 市町村名
area	String 住所1

郵便番号APIの呼び出し例

```
var url = 'https://api.anko.education/zipcode?zipcode=100-0013';  
fetch(url)  
  .then(function(response) {  
    return response.json();  
  })  
  .then(function(address) {  
    document.write(address.pref + address.city + address.area);  
  });
```

変数にWebAPIのアドレスを格納

WebAPIをfetchで呼び出し

成功結果からJSONだけを抽出

JSONの値を変数addressとして取得

JSONの値を画面に書き出し

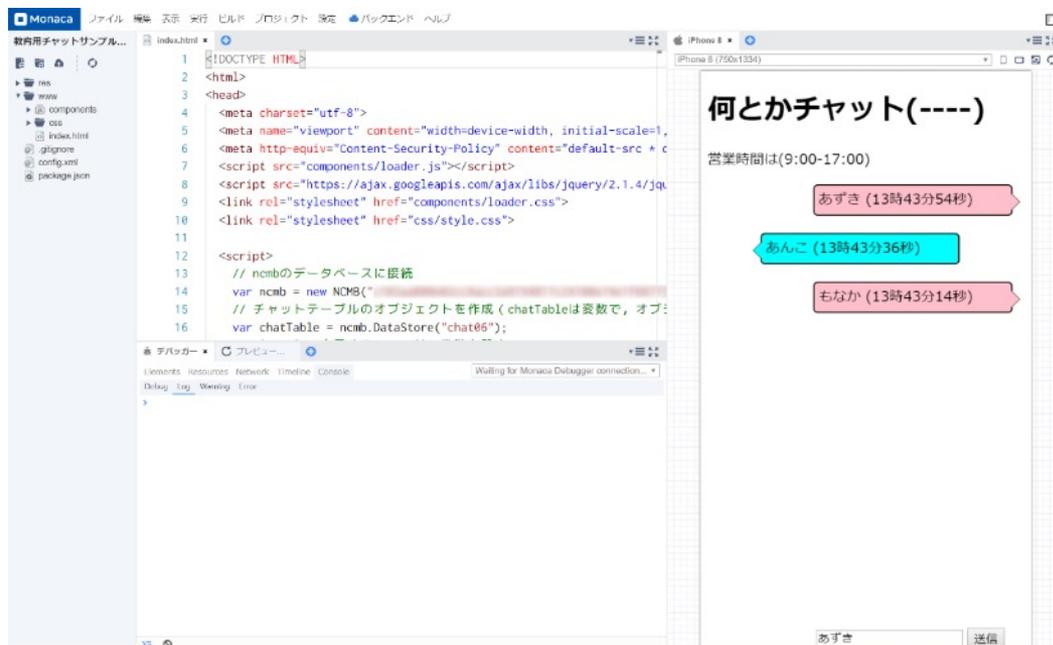
※高等学校情報科「情報Ⅰ」教員研修用教材 3章 P125を参考に作成

Monaca利用事例の紹介

【事例】大分大学教育学部附属中学校

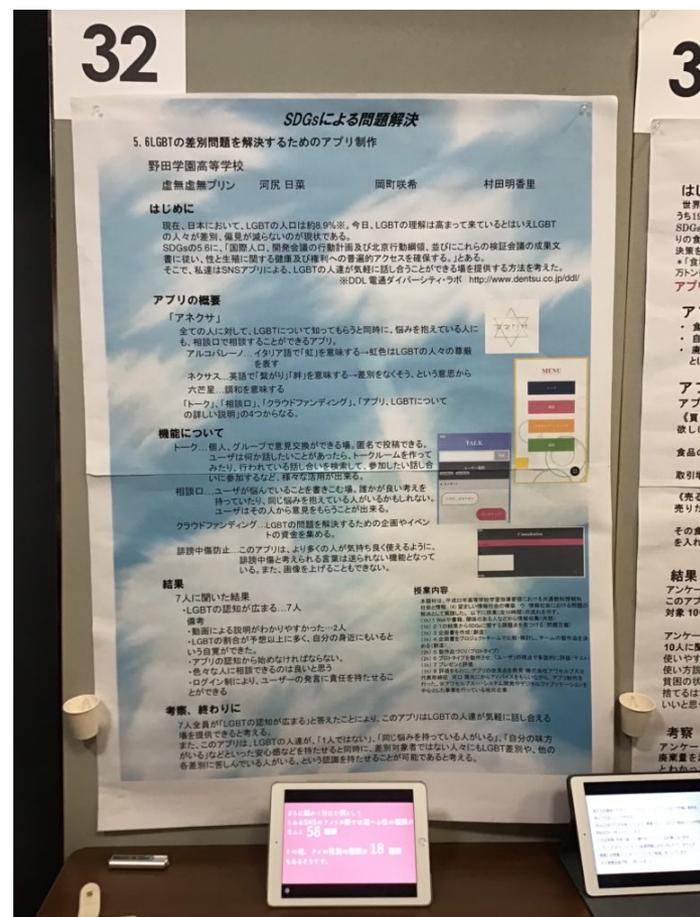
Moancaを利用したチャットアプリの制作

- 技術・家庭科の公開授業で採用
- クラウドデータベースを活用



【事例】野田学園高等学校

- 対象：1年生
- 科目：社会と情報
- 時間：全10時間
- 単元名：SDGsに関する問題を解決するためのアプリ開発
- 実施校：野田学園高等学校
- 教諭名：天川 勇二



【事例】同志社中学校・高等学校

生徒作品例

打数 単打

犠飛 二塁打

四死球 三塁打

本塁打

打率計算 長打率計算 出塁率計算

打率

長打率

出塁率

[もどる](#)

キロクを作成する

読了日: 年 月 日

題名:

作者:

出版社:

初版年: 年

好きなキャラ:

感想・考察:

本棚に並べる

 もどる



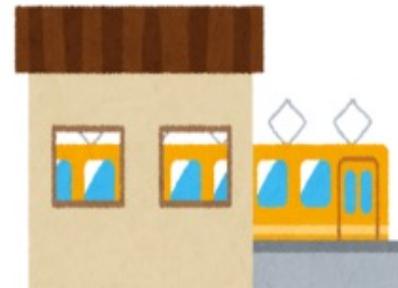
降りる駅を設定してください

駅名:

決定

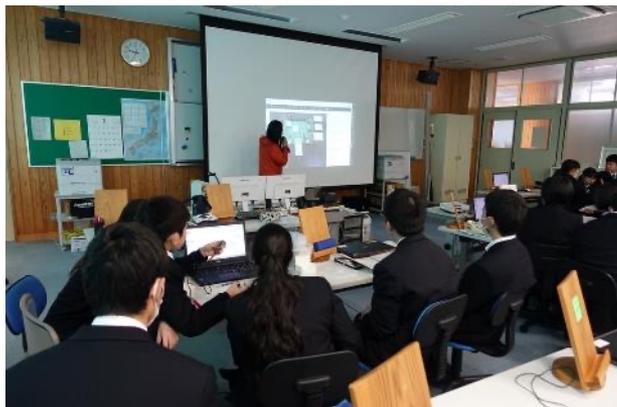
MAP

現在地を取得できませんでした。



【事例】経産省・未来の教室(2018)

- 農業IoTxアプリの授業でMonacaが使われました。



IoT温度・湿度計



【事例】情報処理学会79回全国大会 スタンプラリー

- 名古屋大学 河口研究室が開発



【事例】東京大学国際高等研究所 カブリ数物連携宇宙研究機構

- スマホアプリ開発未経験者が2か月でアプリをリリース! 研究施設のガイドアプリで業務効率化

