

立命館大学情報理工学部 プログラミング講座

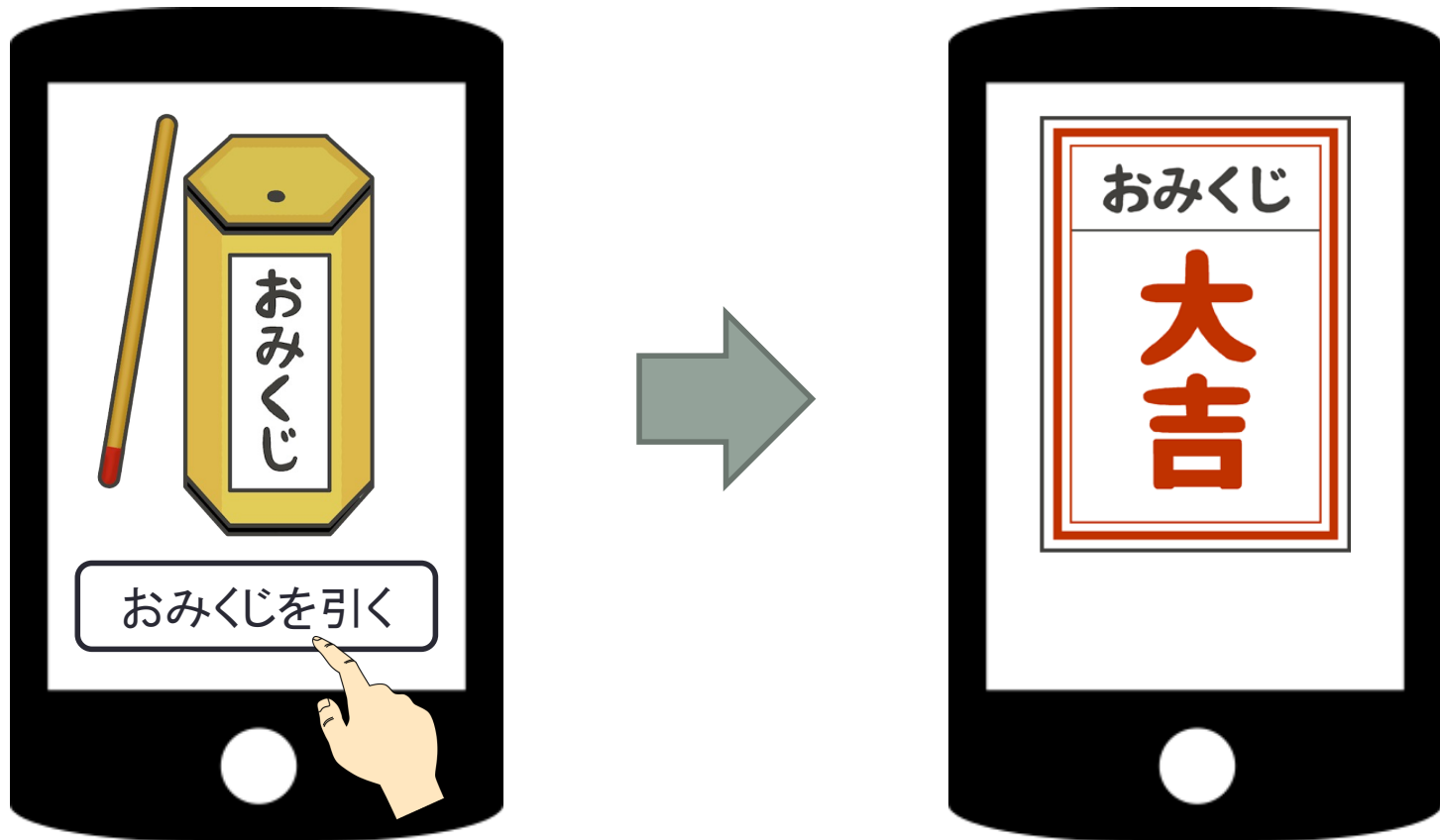
立命館大学 情報理工学部 システムアーキテクトコース
高田 秀志 (htakada@is.ritsumei.ac.jp)



http://www.cm.is.ritsumei.ac.jp/class/omi_ikuei2022/

今日のお題

おみくじアプリの作成



プログラミング言語

世の中にはたくさんのプログラミング言語があります

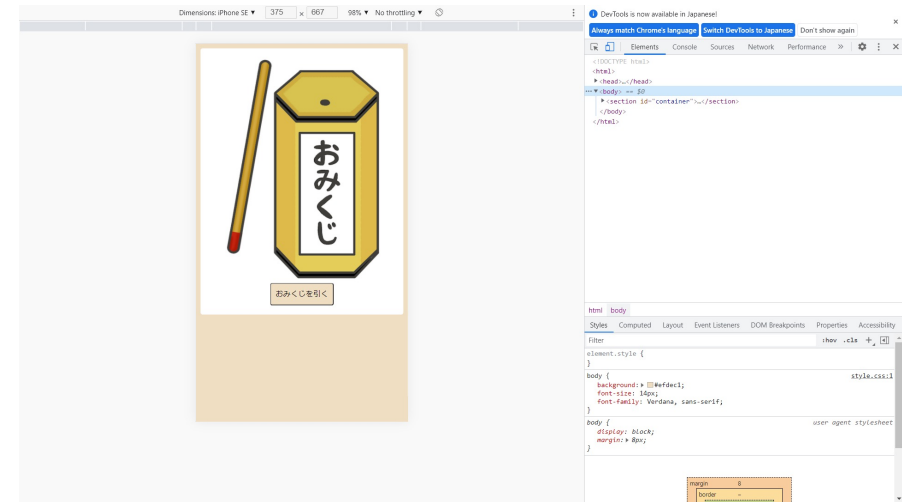
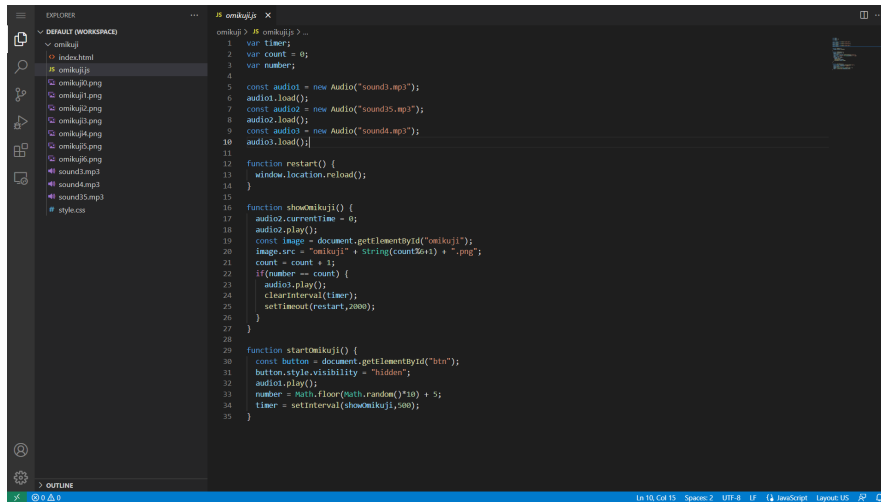
C, C++, C#, Python, Java, JavaScript, Swift, Kotlin, Ruby...

iPhoneアプリはSwift

AndroidアプリはJavaやKotlin

今日は、iPhoneでもAndroidでも動く
HTML5 (JavaScript, HTML, CSS) で
プログラミングをします

Visual Studio Code + Chrome



VSCodeでプログラムを作成
(コーディング)

Chromeで
プログラムを実行

アプリの機能

まずはじめに、アプリ全体でどんな機能が必要なのかを整理する



1. スタート画面の表示
 - 画像ひとつ
 - ボタンひとつ
2. 札を順番に表示
 - 大吉→中吉→小吉・・・
 - 乱数で表示する札の数を決める
3. 仕上げ

スタート画面の表示 (1)

index.html をこのように書き換えて、
画像とボタンを表示する

```
<body>  
  <section>  
    <br/>  
    <button id="button">おみくじを引く</button>  
  </section>  
</body>
```

<body> : 「画面に表示する内容の始まりですよ」という記号

 : 画像を表示する記号 (srcで画像ファイルの名前を指定)

<button> : ボタンを表示する記号

</body> : 「bodyの終わりですよ」という記号

スタート画面の表示 (2)

Chromeで表示してみる

1. 新しいタブを開く
2. index.htmlをChromeにドラッグ&ドロップ
3. デベロッパーツールを開く
4. スマホ表示モードにする



見栄えがしょぼい (;_;)

スタート画面の表示 (3)

スタイルシート (style.cssファイル) で見栄えを変える

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" ...>
  <link rel="stylesheet" href="style.css"> ← この行を追加
</head>
```

本当はスタイルシートもイチから作る必要がありますが、今日は出来上がったものを取りあえず使います。

スタイルシートの一部

```
button {
  background: #efdec1; ← 背景の色
  padding: 8px;
  border-radius: 4px; ← 囲み枠の丸みを調整
  cursor: pointer;
  text-align: center; ← 文字をセンタリング
  color: #000; ← 文字の色を黒に
}
```

など、見栄えに関する細かい設定が書き込まれています

アプリの機能



1. スタート画面の表示
 - 画像ひとつ
 - ボタンひとつ
2. 札を順番に表示
 - 大吉→中吉→小吉・・・
 - 乱数で表示する札の数を決める
3. 仕上げ

できた
(^o^)

札を順番に表示 (1)

omikuji.js に 2 つの「関数 (function)」を作成

```
var timer; ← 「変数(variable)」
```

```
function showOmikuji() {  
  const img = document.getElementById("img");  
  img.src = "omikuji1.png";  
}
```

札を表示するための「関数」
(とりあえず「大吉」を表示)

```
function startOmikuji() {  
  timer = setInterval(showOmikuji, 500);  
}
```

札の表示を始めるための「関数」

関数呼び出し (showOmikuji関数を500ミリ秒毎に呼び出すためのタイマーをセット)

timerは「変数名」

showOmikuji, startOmikujiは「関数名」

札を順番に表示 (2)

index.html から omikuji.js を呼び出す

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" ...>
  <link rel="stylesheet" href="style.css">
  <script src="omikuji.js"></script>
</head>

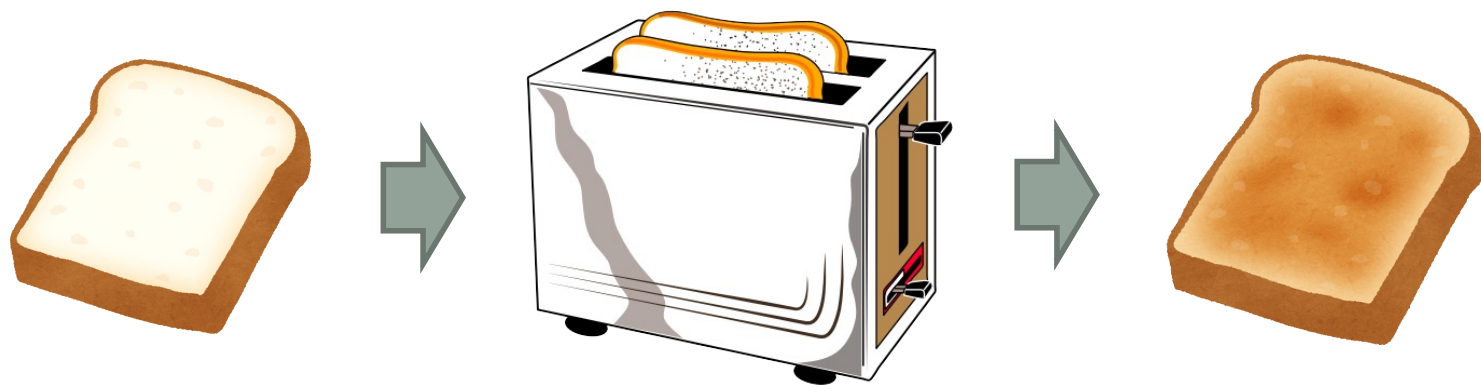
<body>
  <section>
    <br/>
    <button id="button" onclick="startOmikuji();">おみくじを引く</button>
  </section>
</body>
```

← この行を追加

「ボタンがクリックされたら startOmikuji 関数を呼び出してね」という意味

プログラミングの知識 (1)

関数(function)とは「何かをするもの」



```
function toaster() {  
    パンを焼く;  
}
```

- 関数には、「自分で作る関数」と「元から入っている関数」があります
- showOmikujiやstartOmikujiは自分で作った関数、setIntervalは元から入っている関数

プログラミングの知識 (2)

"=" は、「同じ」「等しい」ではなく代入

代入とは「～（左辺）を～（右辺）にする」というイメージ

```
img.src = "omikuj1.png";
```

「img.src を omikuj1.png にする」という意味



「大吉」の画像が入ったファイルの名前

img.src は、「表示する画像の名前（ファイル名など）」を表します

プログラミングの知識 (3)

その他、とりあえず気をつけておくべきこと

- プログラム中の文字は基本的に半角
(文字列として日本語を使う場合は別)
- 開き括弧から閉じ括弧までが一つの単位
括弧を開くたびに、次の行から字下げをする

```
function toaster() {  
    heat();  
}
```

開き括弧

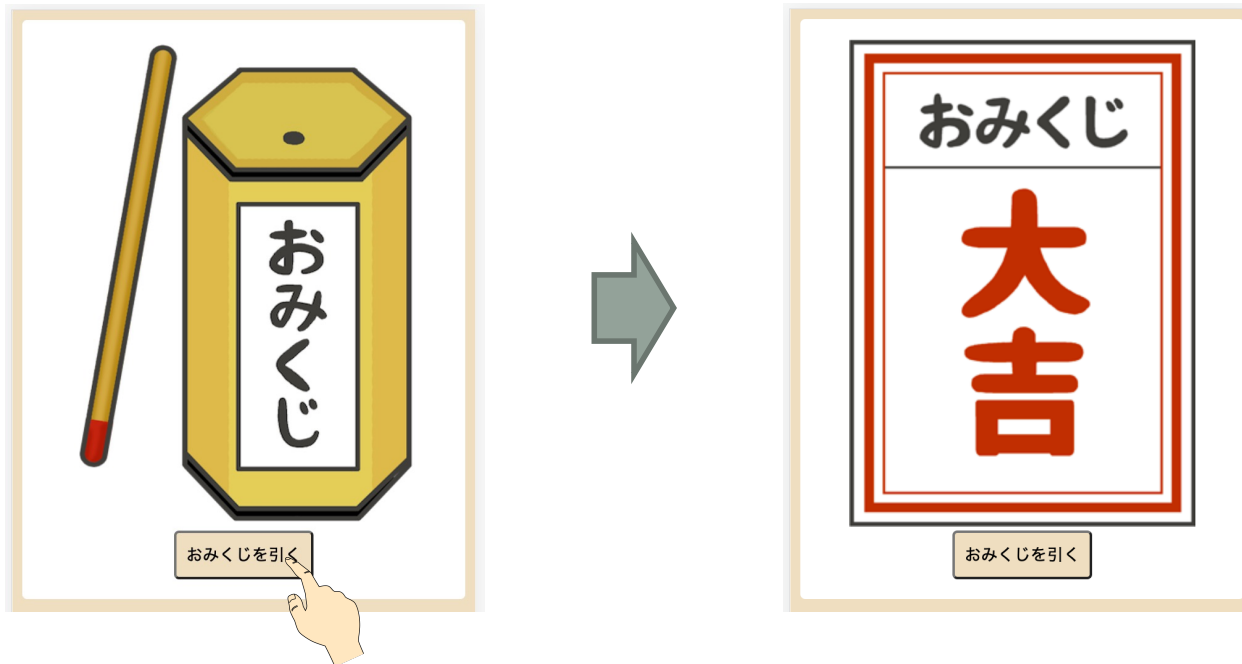
タブキーで字下げ（インデント）を入れる
コンピュータの処理には影響しませんが、人間にとって見やすくなります

閉じ括弧

- 1文字でも間違いがあると全く動きません
(逆に言えば、1文字直すだけで完璧に動き出すこともあります)

ここまでの確認

- index.html と style.css で画像とボタンを表示
- omikuji.js に、タイマーをセットする関数と、大吉を表示する関数を追加
- index.html から omikuji.js に書いた関数を呼び出し



使って楽しいアプリにするには？

- 「大吉」 → 「中吉」 → 「小吉」 → 「末吉」 → 「吉」 → 「凶」 → 「大吉」 → . . . のように順番に表示する

```
img.src = "omikuji1.png";
```

この部分で代入しているファイル名を omikuji1.png → omikuji2.png → omikuji3.png ... のように順番に変えていけばよい

- 順番に表示している札をどこで止めるかは、ランダムに（乱数で）決める

5～14までの整数をランダムに発生させる式

```
Math.floor(Math.random()*10) + 5;
```

- Math.random() は 0以上1未満の小数をランダムに発生させる関数
- Math.floor() は小数点以下を切り捨てる関数

プログラミングの知識 (4)

変数は何かを覚えておくためのもの

```
var count = 0;
```

「札を表示しているのは何回目か」を覚えておくために、
count という変数を作って、それを 0 にするという意味

変数は計算に使える

```
count = count + 1;
```

「count を count + 1 にする」という意味（代入）

（この文を実行すると、count が 0 のときは count が 1 になり、
count が 1 のときは count が 2 になる）

札を順番に表示 (3)

変数 `count` を使って、表示する札を順番に変えていく

```
var count = 0; ← 変数を作成
```

```
function showOmikuji() {  
  const img = document.getElementById("img");  
  img.src = "omikuji" + (count % 6 + 1) + ".png"; ← 変数の値に合わせて  
  count = count + 1; ← ファイル名を設定  
} ← count に 1 を足す
```

- `count % 6 + 1` は「count を 6 で割った余りに 1 を足す」を表す
(count が 0 の時は 1、count が 1 の時は 2、...、count が 8 の時は 3)
- `"omikuji" + (count%6 + 1) + ".png"` は「文字列の連結」
(count%6 + 1 の部分が 3 の場合は omikuji3.png (小吉の札) になる)

この段階では永遠におみくじが止まらない...

プログラミングの知識 (5)

条件により処理を分ける (if文)

```
if(雨が降っていたら) {  
    傘を持つ;  
    長靴を履く;  
}  
else {  
    帽子をかぶる;  
    運動靴を履く;  
}
```

<pre>if: もし else: でなければ</pre>

```
if(お腹が空いていたら) {  
    パンを食べる;  
}
```

elseの部分が無い
場合もあります

札を順番に表示 (4)

乱数で発生させた回数分だけ札を表示したら止める

- 乱数で発生させた回数を覚えておく変数を作成

```
var number;
```

- 回数を乱数で設定

```
function startOmikuji() {  
  number = Math.floor(Math.random()*10) + 5;  
  timer = setInterval(showOmikuji,500);  
}
```

← 5～14の乱数を発生させて、
変数 number で覚えておく

- 回数に達したらタイマーをクリア

```
function showOmikuji() {  
  ...  
  count = count + 1;  
  if(count == number) {  
    clearInterval(timer);  
  }  
}
```

} count と number が等しい場合は、
clearInterval関数を呼び出す

プログラミングの知識 (6)

if文の条件にある `==` は「等しい」(同じ) という意味

<code>a == b</code>	aとbが等しい
<code>a >= b</code>	aがbより大きいか等しい
<code>a > b</code>	aがbより大きい
<code>a <= b</code>	aがbより小さいか等しい
<code>a < b</code>	aがbより小さい
<code>a != b</code>	aがbと等しくない

アプリの機能



1. スタート画面の表示

- 画像ひとつ
- ボタンひとつ

できた
(^o^)

2. 札を順番に表示

- 大吉→中吉→小吉・・・
- 乱数で表示する札の数を決める

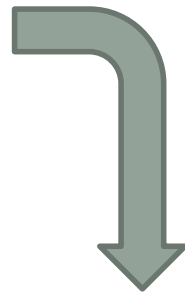
できた
(^o^)

3. 仕上げ

(補足) プログラミングの知識 (7)

同じ処理を繰り返す (for文)

```
月曜日は6時に起きる;  
火曜日は6時に起きる;  
水曜日は6時に起きる;  
木曜日は6時に起きる;  
金曜日は6時に起きる;  
土曜日は10時まで寝る;  
日曜日は10時まで寝る;
```



繰り返し部分を一つにまとめる
繰り返しの中で異なる部分を変数にする

```
for(let day of ["月", "火", "水", "木", "金"]) {  
    day曜日は6時に起きる;  
}  
for(let day of ["土", "日"]) {  
    day曜日は10時まで寝る;  
}
```

曜日を day という変数にしています

デバッグ

2回ボタンを押すと...



ボタンが押されたら、ボタンを非表示にする

```
const button = document.getElementById("button");  
button.style.visibility = "hidden";
```

これまで作ってきたプログラムのどこに追加すればよいか、考えて下さい。

音を鳴らす

1. ネットでフリーの音源を見つけて、音源ファイルをプログラムと同じフォルダに追加

<https://soundeffect-lab.info/sound/button/>
など

2. 音を鳴らすプログラムはこんな感じ

```
const sound = new Audio('sound1.mp3'); // sound1.mp3 は音源ファイルの名前
sound.play();
```

- 「ボタンを押した時」「札を表示した時」「最後の札を表示した時」にそれぞれ別の音を鳴らしてみる
- 出た札によって音を変えてみる（大吉は明るい音、凶は暗い音など）

Monacaの紹介

Webで使える統合開発環境 (IDE)



パソコン上でプログラムを作成
(コーディング)

スマホで
プログラムを実行

今回の「おみくじアプリ」をほぼそのまま作れます

Monacaのセットアップ

- Monacaのページを開く
<https://ja.monaca.io/>
 - 「アカウント新規作成」
 - 「新しいプロジェクトを作る」で
「サンプルアプリケーション」のどれかを選ぶ
 - アプリの名前を選択して「クラウドIDEで開く」を押すと
プログラムの編集画面が開く
-
- スマホに「Monaca」アプリをダウンロードする
 - 自分のアカウントでログインする
 - アプリの名前を選択すると、プログラムを実行できる
(無料アカウントで作れるアプリは3つまで)