

Quizアプリ

Contents

- 配列
- 配列の練習
- 繰り返し処理(for)
- 配列と繰り返し処理の組み合わせ
- クイズアプリ初級編
- クイズアプリ中級編
- 課題発表→P2Pタイム

rules...

- 授業中は常にエディタを起動！
- 考えたことや感じたことはslackのガヤチャンネルでガンガン発信！
- 質問はslackへ！ 他の人の質問にも目を通そう！（同じ質問があるかも）
- 演習時，できた人はスクショなどslackに貼ってアウトプット！
- まずは打ち間違いを疑おう！
 - `{'";` など
- 書いたら保存しよう！（よく忘れる！）
 - `command + s`
 - `ctrl + s`

本日のゴール

- 配列に慣れる
- 繰り返し処理に慣れる
- 上記を用いてクイズアプリを実装！

配列

配列とは？？（ハマリポイントその2）

```
// 配列 (array)  
//   - 複数の値に順番をつけてまとめて扱う方法. 奥（闇）が深い.  
//   - 順番を「インデックス」と呼ぶ. 「0」からスタート！
```

```
// 例
```

```
<script>  
  const array = ['大吉', '中吉', '小吉', '凶', '大凶'];  
  alert(array[0]); // 大吉  
</script>
```



配列とは？？(ハマリポイントその2)

// 配列の定義

```
const list01 = ['月', '火', '水', '木', '金', '土', '日'];  
const list02 = new Array('月', '火', '水', '木', '金', '土', '日');  
console.log(list01);
```

// 配列の長さ

```
console.log(list01.length);    // 7
```

// 取り出し方

```
console.log(list01[0]);        // 月  
console.log(list01[2]);        // 水
```

配列と繰り返し処理

```
// 繰り返し処理で配列の中身を順番に表示

// 配列の定義
const list01 = ['月', '火', '水', '木', '金', '土', '日'];

// 配列の最後の要素まで順番に表示
for(let i = 0; i < list01.length; i++){
  // 「i」は0スタートで順番に1ずつ増加
  // i=0でconsole.log()実行 → i=1でconsole.log()実行 → ...
  console.log(list01[i]);
}
```


繰り返し処理を用いたブラウザ表示

```
// 配列の中身をブラウザに表示

// 配列の定義
const list01 = ['月', '火', '水', '木', '金', '土', '日'];
$('#btn').on('click', function(){
    const array = [];           // 空の配列を作成
    for(var i = 0; i < list01.length; i++){
        // 「.push()」を使用して「array」の配列にデータを追加
        array.push('<p>' + list01[i] + '</p>');
    }
    $('#echo').html(array);     // ブラウザに表示
});
```

よく出てくる処理

```
const list01 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

// 先頭に追加
list01.unshift(0);      // [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
// 末尾に追加
list01.push(11);        // [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
// 先頭を削除
list01.shift();          // [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
// 末尾を削除
list01.pop();            // [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
// 反転
list01.reverse();        // [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

繰り返し処理 (for文)

繰り返し処理とは？？

```
// - 同じ処理を指定した回数実行するやり方
//   - 同じ処理を100回実行したいときに100回書くのか．．？
//   - 配列と相性が良い！

// - 書き方
for(開始条件； 繰り返しを続ける条件； 回数のカウンターの仕方){
    実際に繰り返したい処理    // {}内の処理が繰り返し実行される
}

// - 「i」の値を使って繰り返しをコントロールする．
// - 条件をセミコロン「;」で区切る点に注意！！
```

繰り返し処理とは？？

```
// 例
for(let i=0; i < 100; i++){    // 「i=0で実行」「i=1で実行」...の繰り返し
    console.log(i);           // {}内の処理が繰り返し実行される
}

// 上のコードは以下と同じ意味
console.log(0);    // 「let i=0」なので0からスタート
console.log(1);    // 「i++」なのでiは1ずつ増えて実行される
console.log(2);
// ...
console.log(98);
console.log(99);   // 「i<100」なので99まで実行したら終了
```

配列と組み合わせる

```
// 例
const array = ["G's", "ACADEMY", "FUKUOKA"];

for(let i=0; i < array.length; i++){
  console.log(array[i]); // arrayの0番目, 1番目, 2番目が順に出力される
}

// 上のコードは以下と同じ意味
console.log(array[0]);
console.log(array[1]);
console.log(array[2]); // 「i < array.length-1」なのでi=2で終了
```

配列と組み合わせる

```
// 例：年を選択できる<select>を作成する
const range = [1901, 2000]; // 開始年と終了年の配列の定義

let option = ''; // 出力用の変数を用意

for(let i = range[0]; i < range[1]; i++){ // 開始年から終了年までループ
    option += '<option>' + i + '</option>';
}

$('#hoge').html(option); // ブラウザに出力
```

クイズアプリ初級

今回のアプリケーション

quiz初級編の仕様

- ブラウザ上に問題と選択肢を表示.
- 選択肢をクリックしたら正解かどうかを判定.
- `alert()`で正解不正解を表示. ブラウザ上にも表示.

クイズアプリ01（単数）

コードを書くときに最も重要な要素は??

- 効率
- 経済力
- 時間

選択したら正解不正解を表示！

結果は？

必要な処理

①読み込み時の処理

- 指定した場所に問題文を表示.

②選択肢クリック時の処理

- `input type="radio"`の値を受け取る.
- 正解の値と等しいかどうか判定.
- 正解不正解を出力.

【Point】

「JavaScriptのデータ」を元にして
画面の表示を行う！！

必要な材料の準備

```
// 画面に表示するhtml要素
// 「name」は選択肢共通, 「id」 「value」を個別に設定

<ul>
  <li>
    <input type="radio" name="choices" id="choice00" value="0">
    <label for="choice00">効率</label>
  </li>
  // ...
</ul>
```

ページ読み込み時の処理

```
// 指定した場所に問題文を表示
```

```
// 問題文の定義
```

```
const question = 'コードを書くときに最も重要な要素は??';  
const correctAnswer = 2; //正しい答えを指定  
$('#question').text(question); //要素を指定して表示.
```

選択肢クリック時の処理

```
// クリックした要素の値を受け取り，正解不正解の判定を実行！

$('input[name="choices"]').on('click', function (e) {    //nameで取得
    // eにはイベントの情報が入っている
    if (e.target.value == correctAnswer) { //thisでクリック要素のみ取得
        alert('OK!!');                    // 正解の場合
    } else {
        alert('NG!!');                    // 不正解の場合
    }
});
```

練習！

クイズ初級編を実装しよう！

- まずは問題文を表示
- 選んだ選択肢を特定
- 条件分岐で正解不正解を表示！！

クイズアプリ中級

中級編の仕様

quiz中級編の仕様

- ブラウザ上に問題と選択肢を表示.
- 選択肢をクリックしたら正解かどうかを判定.
- `alert()`で正解不正解を表示. ブラウザ上にも表示.
- 問題数を3問にし, 回答したら次の問題を表示.

クイズアプリ02（複数）

第1問

コードを書くときに最も重要な要素は??

- 効率
- 経済力
- 時間

選択したら正解不正解を表示！
&次の問題へ移動

結果は？

中級編に必要な処理

①読み込み時の処理

- 指定した場所に問題文と選択肢を表示.
- 今何問目かのカウント準備.

②選択肢クリック時の処理

- `input type="radio"`の値を受け取る.
- 正解の値と等しいかどうか判定.
- 正解不正解を出力.
- 今何問目かのカウントを増やす.
- 次の問題を表示する.

【Point】

「JavaScriptの配列」を元にして
画面の表示を行う！！

材料の準備

```
// 問題文, 選択肢, 正解を定義
```

```
const questions = [  
  'コードを書くときに最も重要な要素は??',  
  '課題をつくるときに大事なことは??',  
  'コードを書くのがしんどくなったときはどうする??'  
];  
  
const choices = [  
  ['効率', '経済力', '時間'],  
  ['斬新なアイデア', 'コミットする時間', '洗練されたコード'],  
  ['さらにコードを書く', 'よく寝る', 'ビールを飲む']  
];  
  
const correctAnswer = [2, 1, 0];
```

読み込み時の処理

// カウント用変数と問題文を表示する処理

```
let count = 0; // 何問目かをカウントする変数
let correctCount = 0; // 正解数をカウントする変数
```

// 最初の問題と選択肢を表示

```
$('#question').text(questions[count]); // 問題文の表示
$('#choice00_label').text(choices[count][0]); // 選択肢の表示
$('#choice01_label').text(choices[count][1]); // 選択肢の表示
$('#choice02_label').text(choices[count][2]); // 選択肢の表示
```

選択肢クリック時の処理

```
// 正解不正解の判定と次の問題を表示する処理

$('input[name="choices"]').on('click', function (e) {
  // ...正解不正解の判定（初級と同じ）
  count++;
  // ↑今何問目かのカウントを増やす. 「i++」は「i = i + 1」の意味
  $('#question').text(questions[count]);           // 次の問題を表示
  $('#choice00_label').text(choices[count][0]);    // 次の選択肢を表示
  $('#choice01_label').text(choices[count][1]);    // 次の選択肢を表示
  $('#choice02_label').text(choices[count][2]);    // 次の選択肢を表示
});
```

課題

課題

アップグレード版クイズアプリを作成！

- 問題数を増やす.
- クイズの今何問目かを表示.
- 6問まで終了したら終了の判定をする.
- 終了時に正解数を表示.
- その他, 配列や繰り返しを使用すれば何でもOK!
- todoリスト, 性格診断, tinderみたいなやつ, etc...
- 正直にクイズつくる人はいないだろう(予想)

締切は次回授業前

やばいいい . . . (´;ω;´)

完全に詰んだ．．．という方は

写経

※ 写経とは ※

誰かが書いた「動くコード」をひたすら書き写すこと

```
1  <!DOCTYPE html>
2  <html lang="ja">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>quizアプリ02</title>
9      <style>
10         .header,
11         .footer {
12             width: 100%;
13             background: navy;
14             color: whitesmoke;
15             border: 1px solid black;
16             box-sizing: border-box;
17             text-align: center;
18         }
19
```

```
19
20     main {
21         font-size: 24px;
22         text-align: center;
23     }
24
25     form {
26         display: flex;
27         flex-direction: column;
28     }
29 </style>
30 </head>
31
```

```
32 <body>
33   <header class="header">
34     <h1>クイズアプリ02（複数）</h1>
35   </header>
36   <main>
37     <p id="count">第1問</p>
38     <p id="question"></p>
39     <form action="">
40       <label for="choice00">
41         <input type="radio" name="choices" id="choice00" value="0">
42         <span id="choice00_label">効率</span>
43       </label>
44       <label for="choice01">
45         <input type="radio" name="choices" id="choice01" value="1">
46         <span id="choice01_label">経済力</span>
47       </label>
48       <label for="choice02">
49         <input type="radio" name="choices" id="choice02" value="2">
50         <span id="choice02_label">時間</span>
51       </label>
52     </form>
53   </main>
54   <footer class="footer">
55     <h1 id="result">結果は？</h1>
56   </footer>
57
```

```
58 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
59 <script>
60
61 $(function(){
62
63     // 問題の定義
64     const questions = [
65         'コードを書くときに最も重要な要素は??',
66         '課題をつくるときに大事なことは??',
67         'コードを書くのがしんどくなったときはどうする??'];
68     // 選択肢の定義
69     const choices = [
70         ['効率', '経済力', '時間'],
71         ['面白いアイデア', 'かける時間', '洗練されたコード'],
72         ['さらにコードを書く', '寝る', '飲み会に行く']
73     ];
74     // 正解の定義
75     const answer = [2, 1, 0];
76     // カウント用変数
77     let count = 0;
78     let correct = 0;
79
```



```
80 // 最初の問題と選択肢を表示
81 $('#question').text(questions[count]);
82 $('#choice00_label').text(choices[count][0]);
83 $('#choice01_label').text(choices[count][1]);
84 $('#choice02_label').text(choices[count][2]);
85
86 // クリック時に正誤判定&次の問題の内容を表示
87 $('input[name="choices"]').on('click', function() {
88     if ($(this).val() == answer[count]) {
89         // 正解のときの処理
90         alert('OK!!');
91         $('#result').text('OK!!');
92         // 正解数をカウント
93         correct++;
94     } else {
95         // 誤答時の処理
96         alert('NG!!');
97         $('#result').text('NG!!');
98     }
99     // カウントを増やす
100    count++;
```

```

102     if (count < questions.length) {
103         // 次の問題を表示
104         $('#count').text('第' + (Number(count) + 1) + '問');
105         $('#question').text(questions[count]);
106         $('#choice00_label').text(choices[count][0]);
107         $('#choice01_label').text(choices[count][1]);
108         $('#choice02_label').text(choices[count][2]);
109     } else {
110         // 全部終わったら終了で正解数を表示
111         alert('終了!');
112         alert(correct + '問正解!!');
113     }
114 });
115
116 });
117 </script>
118 </body>
119
120 </html>

```

「写経」はあくまでヒント！
「オリジナルの作品」で力がつく！

P2Pタイム

まずはチーム内で解決を目指す！

訊かれた人は苦し紛れでも応える！！