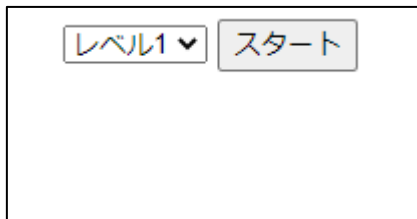


2022 年前期 JavaScript 実習課題

課題はクイズゲームです。下図のような3つの画面があり、「難易度(レベル)の選択」→「表示される英単語の意味を解答」→「正解率を計算／表示」という流れでゲームが進行します。

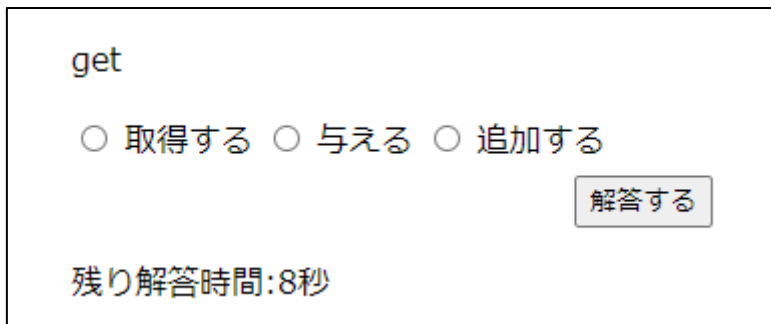
■ ゲームの流れ

開始画面



開始画面のスクリーンショット。左側に「レベル1」のプルダウンメニューがあり、右側に「スタート」ボタンがあります。

プレイ画面



プレイ画面のスクリーンショット。単語「get」が表示され、その下に「○ 取得する ○ 与える ○ 追加する」という3つの選択肢があります。右側には「解答する」ボタンがあります。下部には「残り解答時間:8秒」と表示されています。

終了画面

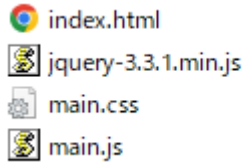


終了画面のスクリーンショット。中央には「ゲーム終了」と大きく表示されています。その下に「正解率: 0%」と表示されています。最下部には「開始画面に戻る」ボタンがあります。

■ 仕様

1. プレイヤーはゲームの難易度(レベル1～3)を選択する。
2. 選択した難易度によって適切なクイズを表示する(難易度によって問題内容や数が変わる)。
3. 1問あたりの解答時間を10秒とし、時間が経過した場合は次の問題へ遷移する。
4. 全ての問題を終わったら正解率を表示する。

■ 基礎となるファイル



※ Teams のファイルに「kadai」というフォルダにて配布しています。

■ クイズデータの取得

クイズに関するデータを ajax で取得し、コンソールに表示するコードです。

```
$(function(){
  var quizData;
  $(document).ready(function(){
    $.ajax({
      url: "https://fieldslong.jp/kadai/",
      dataType: "json"
    }).done(function(data){
      quizData = data;
    });
  });
  $(document).ajaxComplete(function() {
    start();
  });
  function start() {
    console.log(quizData);
  }
});
```

ajax 通信の完了後に処理を開始するためには非同期処理を考慮しなければなりません。この非同期処理については授業で取り上げていないので、上記のコードをそのまま利用して JSON データを取得してください。

■ クイズデータの構造

1. クイズの JSON データは、クイズの難易度ごとにデータ階層が分かれています。
2. 各難易度を示す階層 (level1～level3) の中には、プレイヤーに表示するクイズに関するデータが入っています。
3. Step という接頭辞のキーを持つデータがクイズとして出題される1問あたりのデータに該当し、「問題の内容」「解答の選択肢」「正解の解答」が入っています。

プロパティ名	意味
word	問題の文字列(英単語)
choices	解答の選択肢文字列(日本語訳候補)の配列
answer	正解の文字列(正しい日本語訳)

■ Index.html の構造と表示位置

```
<body>
  <div id="app"></div>
  <script src="./jquery-3.3.1.min.js"></script>
  <script src="./main.js"></script>
</body>
```

1. 「開始画面」「プレイ画面」「終了画面」はすべて<div id="app"></div>の子要素として表示させる。
2. 参考 JavaScript
`document.getElementById('app').appendChild(ele);`
3. 参考 jQuery
`$('#app').append(ele);`

■ 実装手順(参考)

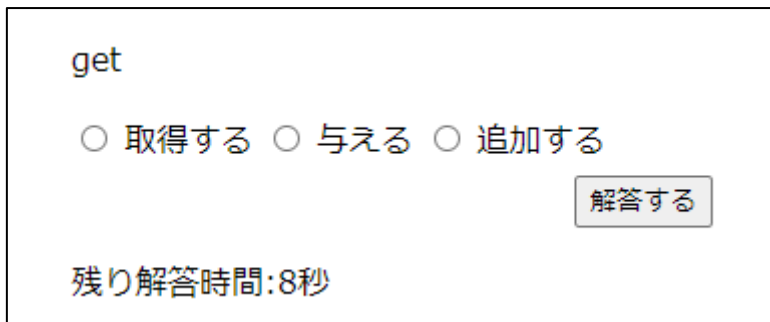
1. クイズデータを取得する(上記コードをそのまま使用する)。
2. 開始画面に、難易度選択セレクトボックスを表示させる。
3. 難易度選択イベントとスタートボタンのクリックイベントを実装する。
4. スタートボタンのクリックイベント後にプレイ画面を表示させる。
5. 選択した難易度はプレイ中に保持するように処理する。
6. 設問を表示し、問題の数に応じて設問を繰り返し表示させる。
7. 解答に制限時間を設定するため、タイマー処理を作成する。
8. クイズの正解率を集計するため、解答結果を保持するよう処理する。
9. 最後の設問の後は、終了画面を表示させる。
10. 正解率の計算処理を実装し、その結果を画面に表示させる。
11. 開始画面へ戻るためのボタンを設置する。

■ 開始画面 HTML(参考)



```
<div>
<select name="selectlevel">
<option value="level1">レベル 1</option>
<option value="level2">レベル 2</option>
<option value="level3">レベル 3</option>
</select>
<button class="startbtn">スタート</button>
</div>
```

■ プレイ画面 HTML(参考)



```
<div class="question">
<p>get</p>
<div>
<label><input type="radio" name="choice" value="取得する">取得する</label>
<label><input type="radio" name="choice" value="与える">与える</label>
<label><input type="radio" name="choice" value="追加する">追加する</label>
</div>
<div class="actions">
<button class="nextbtn">解答する</button>
</div>
<p class="sec">残り解答時間:8 秒</p>
</div>
```

■ 終了画面 HTML(参考)



```
<div class="results">
<h2>ゲーム終了</h2>
<p>正解率: 0%</p>
<button class="resetbtn">開始画面に戻る</button>
</div>
```

■ 配点(参考:変更の可能性あり)

1. 全機能実装完了:60 点
2. タイマーなし:-5 点
3. 正解率計算なし:-5 点
4. 各解答の正誤判定なし:-5 点
5. 問題数に応じた設問の繰り返し表示なし:-5 点
6. 「開始画面」「プレイ画面」「終了画面」の表示切替できず:-5 点
7. レベルに応じた問題が表示できず:-5 点
8. 「終了画面」表示できず:-10 点
9. 「プレイ画面」表示できず:-10 点
10. 「開始画面」表示できず:-10 点