

イントロクイズのプログラムコードリスト

一つのフォルダ内に、[index.html](#), [script.js](#), [style.css](#), [list.js](#) および [music](#) というフォルダを入れます。music というフォルダ内にはイントロの音源の mp3 を入れます。

それらがすべてできたら、index.html をダブルクリックすればイントロクイズが始まります。

(コピペする場合にページ数が入っているとじゃまをするので入れていません。)

index.html

タイトルを変更する場合は赤い文字の部分を変えます。

以下のコードをコピーしてメモ帳に貼り付け、ファイル名を [index.html](#) として保存してください。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>昭和の歌 イン트로クイズ</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <div id="quiz-view">
      <header>
        <h1>🎵 昭和の歌 イン트로クイズ 🎵</h1>
      </header>
      <div id="quiz-info">
        <h2>第 <span id="current-question">1</span> 問</h2>
      </div>
      <div id="button-group">
        <button id="play10Button" class="play-button" data-duration="10">1 回目 (10 秒)</button>
        <button id="play18Button" class="play-button" data-duration="18">2 回目 (18 秒)</button>
        <button id="playFullButton" class="play-button" data-duration="full">3 回目 (最後まで)</button>
      </div>
      <hr>
      <button id="checkAnswerButton" class="control-button" disabled>答え合わせ</button>
      <div id="message-area">
```

```
<p id="current-track-name">再生ボタンを押してイントロを聴いてみましょう
</p>
</div>
</div>
<div id="answer-view" class="hidden">
  <header>
    <h1>💡 正解発表 💡 </h1>
  </header>
  <div id="answer-details">
    <p class="answer-label">曲名</p>
    <h3 id="answer-song">(曲名が入ります)</h3>
    <p class="answer-label">歌手名</p>
    <h3 id="answer-artist">(歌手名が入ります)</h3>
  </div>
  <hr>
  <button id="nextTrackButton" class="control-button"> 次の問題へ
</button>
</div>
<audio id="audio-player" preload="auto"></audio>
</div>

<script src="list.js"></script>
<script src="script.js"></script>
</body>
</html>
```

script.js

以下のコードをコピーしてメモ帳に貼り付け、ファイル名を **script.js** として保存してください。

```
// --- 設定エリア ---
const musicFolderPath = "music/";

// --- ゲーム状態管理 ---
let initialTrackList = []; // 曲ファイル名のリスト
let answerData = {};       // 曲名と歌手名のデータ
let shuffledTrackList = [];
let currentTrackIndex = 0;
let currentTimeout;

// --- DOM 要素取得 ---
const audioPlayer = document.getElementById('audio-player');
const quizView = document.getElementById('quiz-view');
const answerView = document.getElementById('answer-view');
const playButtons = document.querySelectorAll('.play-button');
const checkAnswerButton = document.getElementById('checkAnswerButton');
const nextTrackButton = document.getElementById('nextTrackButton');
const currentQuestionEl = document.getElementById('current-question');
const messageAreaEl = document.getElementById('current-track-name');
const answerSongEl = document.getElementById('answer-song');
const answerArtistEl = document.getElementById('answer-artist');

/**
 * 1. list.js 内の csvData 文字列を解析してデータを準備する
 */
function loadDataAndStart() {
  try {
    // list.js で定義した csvData を改行ごとに分ける
    const rows = csvData.trim().split(/¥r?¥n/);

    // 1 行目(ヘッダー)を飛ばして 2 行目から処理
    for (let i = 1; i < rows.length; i++) {
      const row = rows[i].split(",");
      if (row.length < 3) continue;

      const num = row[0].trim(); // 曲番号 (例: 01)
      const song = row[1].trim(); // 曲名
      const artist = row[2].trim(); // 歌手名
    }
  }
}
```

```

        const fileName = num + ".mp3";
        initialTrackList.push(fileName);
        answerData[fileName] = { song: song, artist: artist };
    }

    // 解析が終わったらゲームを開始
    initializeGame();

} catch (error) {
    console.error("解析エラー:", error);
    alert("データの読み込みに失敗しました。list.js の形式を確認してください。");
}
}

// --- ユーティリティ関数 ---
function shuffleArray(array) {
    for (let i = array.length - 1; i > 0; i--) {
        const j = Math.floor(Math.random() * (i + 1));
        [array[i], array[j]] = [array[j], array[i]];
    }
    return array;
}

// --- イベントリスナー ---
playButtons.forEach(button => {
    button.addEventListener('click', () => {
        const durationType = button.getAttribute('data-duration');
        if (durationType === 'full') {
            playTrackSegment(Infinity);
        } else {
            playTrackSegment(parseInt(durationType));
        }
    });
});

checkAnswerButton.addEventListener('click', showAnswerScreen);
nextTrackButton.addEventListener('click', moveToNextTrack);

audioPlayer.onended = () => {
    enableButtons();

```

```
    checkAnswerButton.disabled = false;
    messageAreaEl.textContent = "曲の再生が終了しました。答え合わせへどうぞ。";
};
```

```
// --- メインロジック ---
```

```
function initializeGame() {
    if (initialTrackList.length === 0) {
        alert("曲データが見つかりません。");
        return;
    }
    shuffledTrackList = shuffleArray([...initialTrackList]);
    currentTrackIndex = 0;
    loadTrack();
}
```

```
function loadTrack() {
    if (currentTrackIndex >= shuffledTrackList.length) {
        endGame();
        return;
    }
```

```
    const currentTrackName = shuffledTrackList[currentTrackIndex];
    audioPlayer.src = musicFolderPath + currentTrackName;
    audioPlayer.load();
```

```
    clearTimeout(currentTimeout);
    audioPlayer.pause();
    audioPlayer.currentTime = 0;
```

```
    quizView.classList.remove('hidden');
    answerView.classList.add('hidden');
```

```
    currentQuestionEl.textContent = currentTrackIndex + 1;
    messageAreaEl.textContent = "再生ボタンを押してイントロを聴いてみましょう";
    checkAnswerButton.disabled = true;
    enableButtons();
}
```

```
function playTrackSegment(duration) {
    if (currentTrackIndex >= shuffledTrackList.length) return;
```

```

clearTimeout(currentTimeout);
audioPlayer.pause();
audioPlayer.currentTime = 0;

disableButtons();
checkAnswerButton.disabled = true;

audioPlayer.play().then(() => {
  if (duration !== Infinity) {
    messageAreaEl.textContent = `現在、${duration}秒まで再生中です...`;
    currentTimeout = setTimeout(() => {
      audioPlayer.pause();
      enableButtons();
      checkAnswerButton.disabled = false;
      messageAreaEl.textContent = `再生を停止しました。${duration}秒まで聴きました。答え合わせへどうぞ。`;
    }, duration * 1000);
  } else {
    messageAreaEl.textContent = "最後まで再生中です... 曲が終わるまでお待ちください。";
  }
}).catch(error => {
  console.error("再生エラー:", error);
  alert(`音楽ファイルが見つかりません :
music/${shuffledTrackList[currentTrackIndex]}`);
  enableButtons();
  checkAnswerButton.disabled = false;
});
}

```

```

function showAnswerScreen() {
  clearTimeout(currentTimeout);
  audioPlayer.pause();

  const currentTrackFile = shuffledTrackList[currentTrackIndex];
  const answer = answerData[currentTrackFile];

  if (answer) {
    answerSongEl.textContent = answer.song;
    answerArtistEl.textContent = answer.artist;
  }
}

```

```

    }

    quizView.classList.add('hidden');
    answerView.classList.remove('hidden');
}

function moveToNextTrack() {
    currentTrackIndex++;
    if (currentTrackIndex < shuffledTrackList.length) {
        loadTrack();
    } else {
        endGame();
    }
}

function endGame() {
    clearTimeout(currentTimeout);
    audioPlayer.pause();
    quizView.classList.add('hidden');
    answerView.classList.remove('hidden');
    answerSongEl.textContent = "全問終了";
    answerArtistEl.textContent = "お疲れ様でした！";
    nextTrackButton.textContent = "ゲーム終了";
    nextTrackButton.disabled = true;
}

function enableButtons() {
    playButtons.forEach(btn => btn.disabled = false);
}

function disableButtons() {
    playButtons.forEach(btn => btn.disabled = true);
}

// 🚀 実行開始
loadDataAndStart();

```

style.css

以下のコードをコピーしてメモ帳に貼り付け、ファイル名を **style.css** として保存してください。

```
body {
  font-family: 'Arial', sans-serif;
  background-color: #fce4ec; /* 楽しいピンク系の背景 */
  color: #333;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  margin: 0;
  text-align: center;
}

/* ★ 修正箇所: コンテナの幅とパディングを調整し、はみ出しを解消 */
.container {
  background-color: #fff;
  padding: 30px; /* パディングを少し減らす */
  border-radius: 20px;
  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);
  width: 90%; /* 幅を広げ、レスポンスに対応 */
  max-width: 550px; /* 最大幅を少し狭くして余裕を持たせる */
}

header h1 {
  color: #e91e63; /* マゼンタ */
  font-size: 2.5em;
  margin-bottom: 30px;
  border-bottom: 3px solid #f06292;
  padding-bottom: 10px;
}

#quiz-info h2 {
  font-size: 1.8em;
  color: #880e4f;
  margin-bottom: 30px;
}

#button-group {
  display: flex;
```



```
    justify-content: space-around;
    gap: 10px;
    margin-bottom: 25px;
}
```

```
.play-button, .control-button {
    color: white;
    padding: 15px 10px;
    font-size: 1.1em;
    border: none;
    border-radius: 10px;
    cursor: pointer;
    transition: background-color 0.3s;
    box-shadow: 0 4px rgba(0, 0, 0, 0.2);
    width: 30%; /* 3 つ並べるため */
}
```

```
/* 再生ボタンのスタイル */
```

```
#play10Button { background-color: #2196f3; } /* 青 */
#play10Button:active { background-color: #1565c0; }
```

```
#play18Button { background-color: #4caf50; } /* 緑 */
#play18Button:active { background-color: #2e7d32; }
```

```
#playFullButton { background-color: #ff9800; } /* オレンジ */
#playFullButton:active { background-color: #f57c00; }
```

```
/* 答え合わせ/次の曲ボタンのスタイル */
```

```
#checkAnswerButton, #nextTrackButton {
    background-color: #e91e63; /* マゼンタ */
    padding: 15px 30px;
    font-size: 1.5em;
    width: 95%; /* 広く使う */
    margin-top: 10px;
    box-shadow: 0 5px #c2185b;
}
```

```
#checkAnswerButton:active, #nextTrackButton:active {
    background-color: #ad1457;
    box-shadow: 0 2px #c2185b;
    transform: translateY(3px);
}
```

```
}
```

```
/* ボタンが押せない状態のデザイン */
```

```
.play-button:disabled, .control-button:disabled {  
    background-color: #bdbdbd !important;  
    cursor: not-allowed;  
    box-shadow: none;  
    transform: none;  
}
```

```
/* --- メッセージ/表示エリア --- */
```

```
#message-area p {  
    margin-top: 25px;  
    font-size: 1.2em;  
    color: #880e4f;  
    font-weight: bold;  
}
```

```
hr {  
    border: 0;  
    height: 1px;  
    background-image: linear-gradient(to right, rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.3),  
    rgba(0, 0, 0, 0));  
    margin: 20px 0;  
}
```

```
/* --- 画面切り替え用 --- */
```

```
.hidden {  
    display: none !important;  
}
```

```
/* --- 答え合わせ画面のスタイル --- */
```

```
#answer-view {  
    padding: 10px; /* ★ 修正箇所: パディングを減らす */  
}
```

```
#answer-view header h1 {  
    color: #4caf50; /* 緑色で正解を目立たせる */  
    border-bottom: 3px solid #81c784;  
}
```

```
#answer-details {
    margin: 20px 0; /* ★ 修正箇所: マージンを減らす */
    padding: 15px; /* ★ 修正箇所: パディングを減らす */
    background-color: #f1f8e9; /* 薄い緑の背景 */
    border-radius: 10px;
    border: 1px solid #c8e6c9;
}

.answer-label {
    font-size: 1.1em;
    color: #555;
    margin-top: 15px;
    margin-bottom: 3px;
}

/* ★ 修正箇所: 曲名/歌手名のフォントサイズを調整 */
#answer-song, #answer-artist {
    font-size: 2.0em; /* 2.2em から 2.0em へ縮小 */
    color: #333;
    margin: 0;
    word-wrap: break-word; /* 長いテキストがはみ出さないように */
}
```

list.js

以下のコードをコピーしてメモ帳に貼り付け、自分が収録した曲のリストに入れ替え、(すなわち赤い字の部分を削除して新しいリストに書き換え)ファイル名を **list.js** として保存してください。

その際、最後の `;` の「`」が抜けやすいので注意してください。

この list.js を作るには、まず、セクセルで作成・保存したものを「csv utf-8(コンマ区切り)」として保存し、それをメモ帳で開いて貼り付けるのが一番間違いがないです。曲番号が一桁の場合、0 が前に入っていないといけません。

```
// list.js
const csvData = `曲番号,曲名,歌手
01,狙いうち,山本リンダ
02,渚のシンドバッド,ピンクレディ
03,大阪の女,ザピーナッツ
04,高校三年生,舟木一夫
05,群青,谷村新司`;
```