

JavaScript 三目並べ解答例

タグ：JavaScript

2022年 06月 22日

解答例 1 最もシンプルなパターン

- ・特殊なメソッドを使わずに書いたパターン。動けば良いんや・・・
- ・ゲームが終了してもそのまま続行したり、同じ箇所をクリックできてしまう状態。
- ・count変数はゲームの先後手を確認するために利用。
- ・gameArray配列はゲームの現在の状況を管理するための配列。
- ・このgameArray配列を利用して、ゲームの勝者が決まる縦横斜めの8パターンの並びをチェックする。

```
<!DOCTYPE html>
<html lang="ja">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="If=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    .board {
      margin: 100px auto 0 auto;
      width: 600px;
      display: grid;
      grid-template-columns: 1fr 1fr 1fr;
    }

    .board div {
      border: solid 1px black;
      width: 200px;
      height: 200px;
      font-size: 100px;
      text-align: center;
      line-height: 2;
    }
  </style>
</head>

<body>
  <div class="board">
    <div id="0"></div>
    <div id="1"></div>
    <div id="2"></div>
    <div id="3"></div>
    <div id="4"></div>
    <div id="5"></div>
    <div id="6"></div>
    <div id="7"></div>
    <div id="8"></div>
  </div>
</body>
</script>
<script>
  // 現在の手番をチェックする
  let count = 0;

  // 現在のゲームの進行状況を保存する配列
  const gameArray = [0, 1, 2, 3, 4, 5, 6, 7, 8];

  // ゲームの勝ち負けをチェックする関数
  const checkWinner = function (player) {

    // 横のチェック
    if (gameArray[0] === gameArray[1] && gameArray[0] === gameArray[2]) {
      console.log(player + "の勝ち!");
    }
    if (gameArray[3] === gameArray[4] && gameArray[3] === gameArray[5]) {
      console.log(player + "の勝ち!");
    }
    if (gameArray[6] === gameArray[7] && gameArray[6] === gameArray[8]) {
      console.log(player + "の勝ち!");
    }
    }

    // 縦のチェック
    if (gameArray[0] === gameArray[3] && gameArray[0] === gameArray[6]) {
      console.log(player + "の勝ち!");
    }
    if (gameArray[1] === gameArray[4] && gameArray[1] === gameArray[7]) {
      console.log(player + "の勝ち!");
    }
    if (gameArray[2] === gameArray[5] && gameArray[2] === gameArray[8]) {
      console.log(player + "の勝ち!");
    }
    }

    // ナナメのチェック
    if (gameArray[0] === gameArray[4] && gameArray[0] === gameArray[8]) {
      console.log(player + "の勝ち!");
    }
    if (gameArray[2] === gameArray[4] && gameArray[2] === gameArray[6]) {
      console.log(player + "の勝ち!");
    }
    }
  }

  for (let i = 0; i < 9; i++){
    document.getElementById(i).addEventListener('click', function () {

      if (count % 2 === 0) {
        this.textContent = "O";
        gameArray[i] = "O";
        checkWinner("O")
      } else {
        this.textContent = "x";
        gameArray[i] = "x";
        checkWinner("x")
      }
      count++;
    })
  }
</script>
</html>
```

解答例 2 配列のメソッドを多く利用したパターン。

- ・htmlにdivタグを9個作る代わりに、createElement('div')でdiv要素を作成して配置する。
- ・配列のメソッドmap()、every()、includes()を利用。
- ・クリックのたびにgameArray配列を見て、そのマスをクリック済みかを確認する。
- ・flag変数を作成し、ゲームが終了したかを管理させる。
- ・checkArray配列はチェックするべきパターンの配列を8個持った二次元配列。ここからmap()を使うと、内部の配列を一つずつ取り出せる。

```
<!DOCTYPE html>
<html lang="ja">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="If=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    .board {
      margin: 100px auto 0 auto;
      width: 600px;
      display: grid;
      grid-template-columns: 1fr 1fr 1fr;
    }

    .board div {
      border: solid 1px black;
      width: 200px;
      height: 200px;
      font-size: 100px;
      text-align: center;
      line-height: 2;
    }
  </style>
</head>

<body>
  <div class="board">
    </div>
  </body>
</script>
<script>
  // 現在の手番をチェックする
  let count = 0;

  // ゲームが終了したかを表すフラグ
  let flag = false;

  // 現在のゲームの進行状況を保存する配列
  const gameArray = new Array(9).fill(0);

  // 三つの値が一致しているかチェックする関数 booleanを返す
  // 引数として要素が三つの配列を受け取る
  const checkSame = (array) => {
    const a = gameArray[array[0]];
    const b = gameArray[array[1]];
    const c = gameArray[array[2]];
    if (a === b && b === c) {
      return true;
    }
    return false;
  }

  // チェックすべき縦横斜めの組み合わせを記録した配列
  const checkArray = [
    [0, 1, 2], [3, 4, 5], [6, 7, 8],
    [0, 3, 6], [1, 4, 7], [2, 5, 8],
    [0, 4, 8], [2, 4, 6]
  ];

  const checkWinner = (player) => {
    // 各パターンで検証して、一つでも成立していればtrueを返す
    const checkBoard = checkArray.map(array => checkSame(array)).includes(true);

    if (checkBoard) {
      // ゲームが終了したので、flagをtrueにする
      flag = true;
      console.log(player + "の勝ち!");
    }
  }

  const board = document.querySelector('.board')

  // board要素の中にdivを9個作る
  for (let i = 0; i < 9; i++) {
    const masu = document.createElement('div');
    // 作ったdiv要素にイベントを追加
    masu.addEventListener('click', () => {

      // すでにクリックしていないマスであるかつゲームが続いていることを確認
      if (gameArray[i] === 0 && flag === false) {

        if (count % 2 === 0) {
          masu.textContent = "O";
          gameArray[i] = "O";
          checkWinner("O")
        } else {
          masu.textContent = "x";
          gameArray[i] = "x";
          checkWinner("x")
        }
        count++;
      }
    })
    board.appendChild(masu)
  }
</script>
</html>
```