

WEB230 - JavaScript 1

Assignment 6a

Use Events and DOM manipulation to create a playable Tic Tac Toe game. Do the following using JavaScript. Do not modify the HTML file.

When you see this symbol: 🖐️ save and refresh the browser (this should happen automatically if you are using Brackets with Live Preview), then check your code to make sure it is working before you proceed.

1. Create two variables called “playerX” and “playerO”. Set them to the DOM element with the id “X” and “O” respectively.
2. Add the class “current-player” to the “playerX” element. Notice that the style changes on the “Player X” at the top of the page.
3. Create a named function with whatever name you like. (This will be referred to as the “event handler” in further instructions.) In this function add a `console.log()` statement to display a message in the console.
4. Use a for loop to add this function as a `click` event handler to each of the TD elements in the table. 🖐️ Open the console and click a square. You should see your message.
5. In your event handler, add a statement to get the id of the element with the class “current-player” and store it to a variable. Display this value in the `console.log()` you already have.
6. At the end of your event handler, add a statement to switch the “current-player” class from `playerX` to `playerO` or visa-versa. (Remember you can toggle classes.)
7. In your event handler, add a statement to add the class “X-marker” or “O-marker” to the target element depending on who’s turn it is. (the target element is the one that was clicked on. Hint: the event object has a `target` property.) 🖐️ You should see X or O appear when you click on a square.
8. Add an event handler to the “New Game” button to clear all the squares. 🖐️
9. Notice that if you click on a square that is taken by the other player, the marker may turn into a question mark. This is because you have added both “X-marker” and “O-marker” classes to the document. If you used `className` you would be able to take over another players square. Also, if you click on your own square you wasted your turn. Modify the event handler to check to see if there is a class on the element already. (Hint: this is a good use of `className` but there are other ways to do it.) Don’t put a marker or count the turn if the square is already taken. 🖐️
10. Add the code below to the beginning of your program. This will be used to check if there is a winner.

```
// Array of winning combinations  
const wins = [  
  [0, 1, 2],  
  [3, 4, 5],
```

```

    [6, 7, 8],
    [0, 3, 6],
    [1, 4, 7],
    [2, 5, 8],
    [0, 4, 8],
    [2, 4, 6]
];
/*
    The argument to `checkWin` is the class of the player who's turn it is.
    I.E. X-marker or O-marker
    `checkWin` returns true if there is a winner and false if not
*/
function checkWin(player) {
    // `squares` is an list of all the TD elements in the game.
    // Change it to the name of the variable you used
    let grid = Array.from(squares, function(square) {
        return square.classList.contains(player);
    });
    for (let win of wins) {
        if (grid[win[0]] && grid[win[1]] && grid[win[2]]) {
            for (let sq of win) {
                // Add code here to set background color to 'lightgreen'.
                // `sq` is the index of the TD
            }
            return true;
        }
    }
    return false;
}

```

11. Call the `checkWin()` function from the event handler. The `player` argument is “X-marker” or “O-marker”. Change the name of the `squares` variable to match the name you used for your list of `td` elements. 🖐️
12. When there is a winner, set the background colour of the three winning squares to highlight them and open an alert box to say who the winner is. Do this by adding code where the comment is in the `checkWin()` function.
13. In the event handler for the “New Game” button, add code to remove the background colour and reinstate the event handler for the table. 🖐️

Optional items. Not for extra marks but think how satisfying it will be! 😊

1. Stop the game play once there is a winner. You can do this using a global variable set to `false` until there is a winner. Be sure to reset it when you press the “New Game” button.
2. Display a message if the game ends with no winner.

3. Add a `setTimeout()` to the winner alert. Delay it about 50 milliseconds so the board will draw before the alert pops up.