

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Tic-Tac-Toe</title>

  <style>

    body { font-family: Arial, sans-serif; text-align: center; }

    .board {

      display: grid;

      grid-template-columns: repeat(3, 100px);

      grid-template-rows: repeat(3, 100px);

      gap: 5px;

      justify-content: center;

      margin-top: 20px;

    }

    .cell {

      width: 100px;

      height: 100px;

      display: flex;

      align-items: center;

      justify-content: center;

      font-size: 36px;

      font-weight: bold;
```

```
        background-color: red;

        cursor: pointer;
    }

    .x { color: blue; }

    .o { color: green; }

</style>

</head>

<body>

    <h1>Tic-Tac-Toe</h1>

    <div class="board" id="board"></div>

    <h2 id="status"></h2>

    <button onclick="resetGame()">Reset Game</button>

    <script>

        let board = ["", "", "", "", "", "", "", "", ""];

        let currentPlayer = 'X';

        let gameActive = true;

        function createBoard() {

            const boardElement = document.getElementById('board');

            boardElement.innerHTML = "";

            board.forEach((cell, index) => {

                const cellElement = document.createElement('div');

                cellElement.classList.add('cell');
```

```
    cellElement.setAttribute('data-index', index);
    cellElement.innerHTML = cell;
    if (cell === 'X') cellElement.classList.add('x');
    if (cell === 'O') cellElement.classList.add('o');
    cellElement.addEventListener('click', handleMove);
    boardElement.appendChild(cellElement);
  });
}
```

```
function handleMove(event) {
  const index = event.target.getAttribute('data-index');
  if (board[index] || !gameActive) return;
  board[index] = currentPlayer;
  currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
  checkWinner();
  createBoard();
}
```

```
function checkWinner() {
  const winPatterns = [
    [0, 1, 2], [3, 4, 5], [6, 7, 8],
    [0, 3, 6], [1, 4, 7], [2, 5, 8],
    [0, 4, 8], [2, 4, 6]
  ];
  ;
```

```

for (let pattern of winPatterns) {
  const [a, b, c] = pattern;
  if (board[a] && board[a] === board[b] && board[a] === board[c]) {
    document.getElementById('status').innerText = `${board[a]} Wins!`;
    gameActive = false;
    return;
  }
}

if (!board.includes("")) {
  document.getElementById('status').innerText = 'Draw!';
  gameActive = false;
}
}

function resetGame() {
  board = ["", "", "", "", "", "", "", "", ""];
  currentPlayer = 'X';
  gameActive = true;
  document.getElementById('status').innerText = "";
  createBoard();
}

createBoard();
</script>

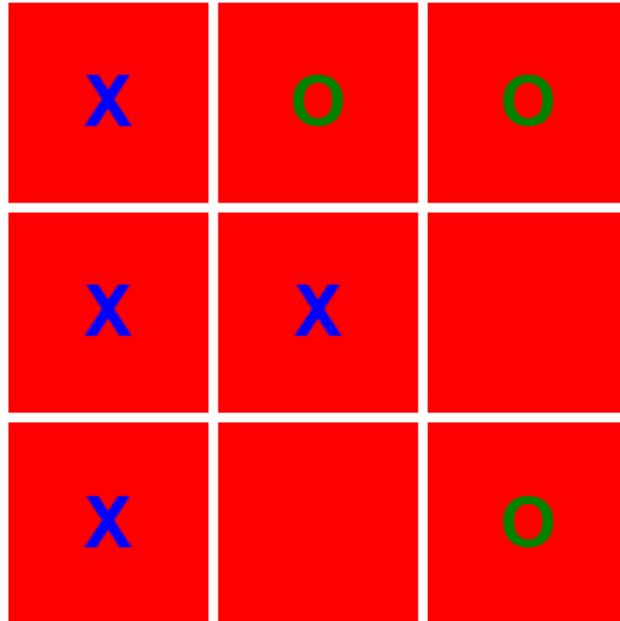
```

```
</body>
```

```
</html>
```

OUTPUT:-

Tic-Tac-Toe



X Wins!

Reset Game