Allgemeine Erklärung

<!DOCTYPE html>: Dieser Deklarationstyp gibt an, dass es sich um eine HTML5-Seite handelt.

<html>...</html>: Dieser Bereich umschließt den gesamten HTML-Inhalt der Seite.

<head>...</head>: Dieser Bereich enthält Metadaten und Verweise auf externe Ressourcen wie CSS und JavaScript.

<title>Tic Tac Toe</title>: Dieser Tag definiert den Titel der Seite, der im Browser-Tab angezeigt wird.

<link rel="stylesheet" type="text/css" href="style.css">: Dieser Tag verbindet die externe CSS-Datei style.css mit der HTML-Seite.

<style>...</style>: Dieser Bereich enthält CSS-Regeln, die direkt in der HTML-Datei definiert sind.

<body>...</body>: Dieser Bereich enthält den sichtbaren Inhalt der Webseite.

<h1>Tic Tac Toe</h1>: Dieser Tag erzeugt eine Überschrift für das Spiel.

<div id="board">...</div>: Dieser Tag definiert den Container für das Tic Tac Toe-Spielfeld.

<div class="row">...</div>: Diese Tags definieren eine Zeile im Spielfeld.

<div class="cell" onclick="makeMove(0, 0)"></div>: Diese Tags definieren ein Zellenfeld im Spielfeld und weisen ihm die Funktion makeMove(row, col) zu, die aufgerufen wird, wenn auf das Feld geklickt wird.

<button onclick="resetBoard()">Neustarten</button>: Dieser Tag definiert einen Button, der die Funktion resetBoard() aufruft, wenn er geklickt wird.

<script>...</script>: Dieser Bereich enthält den JavaScript-Code, der die Logik des Spiels implementiert.

let board = ...: Dieser Code definiert eine 2D-Array-Variable board, die den aktuellen Zustand des Spielfelds speichert.

let currentPlayer = "X";: Diese Variable speichert den aktuellen Spieler (entweder "X" oder "O").

function makeMove(row, col) { ... }: Diese Funktion wird aufgerufen, wenn ein Spieler auf ein Spielfeld klickt. Sie überprüft, ob das Feld leer ist, setzt den Zug des aktuellen Spielers und aktualisiert das Spielfeld im HTML.

function checkWinner() { ... }: Diese Funktion überprüft den aktuellen Zustand des Spielfelds auf einen Gewinner oder ein Unentschieden.

function announceWinner(winner) { ... } und function announceDraw() { ... }: Diese Funktionen zeigen eine Benachrichtigung an, wenn ein Spieler gewinnt oder es zu einem Unentschieden kommt.

function resetBoard() { ... }: Diese Funktion setzt das Spielfeld zurück, indem es alle Zellen leert und das board-Array zurücksetzt.

Die CSS-Regeln definieren das Aussehen des Spielfelds, der Zellen und des Neustarten-Buttons.

Das JavaScript-Code implementiert die Spiellogik. Es überprüft den Zustand des Spielfelds, erkennt Gewinner oder Unentschieden und aktualisiert das Spielfeld im HTML-Dokument.

<h1>Tic Tac Toe</h1>: Dieser Tag erzeugt eine Überschrift "Tic Tac Toe", die über dem Spielfeld angezeigt wird.

<div id="board">...</div>: Dieser Tag definiert den Container für das Spielfeld. Das id-Attribut "board" ermöglicht es, dieses Element im JavaScript-Code leicht zu identifizieren.

<div class="row">...</div>: Diese Tags definieren eine Zeile im Spielfeld. Es gibt insgesamt drei Zeilen im Spielfeld.

<div class="cell" onclick="makeMove(0, 0)"></div>: Diese Tags definieren ein Zellenfeld im Spielfeld. Das class-Attribut "cell" ermöglicht es, dieses Element im CSS-Code leicht zu identifizieren. Das onclick-Attribut weist dem Feld die Funktion makeMove(row, col) zu, die aufgerufen wird, wenn auf das Feld geklickt wird. Die Werte 0, 0 in makeMove(0, 0) geben die Position des Feldes in der Zeile und Spalte an.

HTML Erklärung

<h1>Tic Tac Toe</h1>: Dieser HTML-Tag erzeugt eine Überschrift "Tic Tac Toe", die auf der Webseite angezeigt wird.

<div id="board"> ... </div>: Dieser div-Container mit der id "board" umschließt das gesamte Spielfeld und seine Zellen.

<div class="row"> ... </div>: Diese div-Container mit der Klasse "row" repräsentieren die Zeilen des Spielfelds. Jeder div-Container umschließt eine Gruppe von drei Zellen.

<div class="cell" onclick="makeMove(row, col)"></div>: Diese div-Container mit der Klasse "cell" repräsentieren die einzelnen Zellen des Spielfelds. Jede Zelle ist anklickbar und führt die Funktion "makeMove(row, col)" aus, wobei die Parameter "row" und "col" die Position der Zelle im Spielfeld angeben. Durch Klicken auf eine Zelle wird der Zug des Spielers an dieser Position ausgeführt.

<button onclick="resetBoard()">Neustarten</button>: Dieser Button ermöglicht es dem Benutzer, das Spielfeld zurückzusetzen und das Spiel neu zu starten. Durch Klicken auf den Button wird die Funktion "resetBoard()" ausgeführt.

Javascript Code Erklärung

let board = [...]: Diese Variable board ist ein 2D-Array, das den Zustand des Spielfelds speichert. Jedes Element im Array repräsentiert eine Zelle im Spielfeld und enthält entweder den Wert des aktuellen Spielers ("X" oder "O") oder einen leeren String, wenn das Feld noch nicht gesetzt wurde.

let currentPlayer = "X";: Diese Variable speichert den aktuellen Spieler. Sie wird zu Beginn auf "X" gesetzt.

function makeMove(row, col) { ... }: Diese Funktion wird aufgerufen, wenn ein Spieler auf ein Spielfeld klickt. Sie überprüft, ob das Feld leer ist (board[row][col] === ""). Wenn ja, wird der Zug des aktuellen Spielers auf dem Spielfeld gesetzt (board[row][col] = currentPlayer) und der Text in der entsprechenden Zelle im HTML aktualisiert (document.getElementById("board").children[row].children[col].innerText = currentPlayer). Anschließend wird der aktive Spieler gewechselt, indem currentPlayer auf "O" gesetzt wird, wenn der vorherige Spieler "X" war, oder umgekehrt (currentPlayer = currentPlayer === "X" ? "O" : "X"). Schließlich wird die Funktion checkWinner() aufgerufen, um zu überprüfen, ob ein Spieler gewonnen hat.

function checkWinner() { ... }: Diese Funktion überprüft den aktuellen Zustand des Spielfelds auf einen Gewinner oder ein Unentschieden. Zuerst werden die Reihen überprüft, indem jede Reihe auf Gleichheit der Zellen überprüft wird. Dann werden die Spalten und Diagonalen überprüft. Wenn ein Gewinner gefunden wird, wird die Funktion announceWinner(winner) aufgerufen, wobei der Gewinner als Argument übergeben wird. Wenn es zu einem Unentschieden kommt, wird die Funktion announceDraw() aufgerufen.

function isBoardFull() { ... }: Diese Funktion überprüft, ob das Spielfeld voll ist, indem sie jede Zelle überprüft. Wenn mindestens eine Zelle leer ist, gibt sie false zurück. Andernfalls gibt sie true zurück.

function announceWinner(winner) { ... } und function announceDraw() { ... }: Diese Funktionen zeigen eine Benachrichtigung an, wenn ein Spieler gewinnt oder es zu einem Unentschieden kommt. Sie verwenden setTimeout, um die Benachrichtigung um 100 Millisekunden zu verzögern. Nachdem die Benachrichtigung angezeigt wurde, wird die Funktion resetBoard() aufgerufen, um das Spielfeld zurückzusetzen.

function resetBoard() { ... }: Diese Funktion setzt das Spielfeld zurück, indem sie das board-Array zurücksetzt und den Text in allen Zellen im HTML löscht. Sie verwendet document.getElementsByClassName("cell"), um alle Zellen zu erhalten, und setzt den Text in jeder Zelle auf einen leeren String.

CSS Erklärung

#board { ... }: Dieser Selektor definiert die Regeln für das Spielfeld. Mit display: flex; wird das Spielfeld als Flexbox-Container angezeigt. flex-direction: column; legt die Ausrichtung der Flexbox auf eine vertikale Anordnung fest, sodass die Zeilen des Spielfelds untereinander angeordnet werden. align-items: center; zentriert die Inhalte des Spielfelds vertikal.

.row { ... }: Dieser Selektor definiert die Regeln für die Zeilen des Spielfelds. display: flex; stellt sicher, dass die Zellen in einer Zeile als Flexbox-Elemente angezeigt werden, wodurch sie nebeneinander angeordnet werden.

.cell { ... }: Dieser Selektor definiert die Regeln für die Zellen des Spielfelds. width: 100px; und height: 100px; setzen die Breite und Höhe der Zellen auf 100 Pixel. border: 1px solid black; fügt einen 1-Pixel schwarzen Rahmen um jede Zelle hinzu. display: flex; zeigt den Inhalt der Zelle als Flexbox-Element an. align-items: center; und justify-content: center; zentrieren den Inhalt der Zelle horizontal und vertikal. font-size: 2em; setzt die Schriftgröße auf 2em, was etwa der doppelten Größe der Standard-Schriftgröße entspricht. cursor: pointer; ändert den Cursor beim Überfahren der Zelle in einen Zeiger, um anzuzeigen, dass sie anklickbar ist.

button { ... }: Dieser Selektor definiert die Regeln für die Schaltfläche "Neustarten". margin-top: 20px; fügt einen oberen Abstand von 20 Pixeln zur Schaltfläche hinzu, um sie vom Spielfeld zu trennen.