|  |  |
| --- | --- |
| .textContent Property Setting the .textContent property on an element with a child will remove that child.  Setting .textContent will delete all child nodes and replace them with just the text.   |  | | --- | | <div id=”ElementWithChildren”>  Other Text  <button>  Button that will be deleted if .textContent=”…” is set  </button>  </div> |   To only modify the initial text access the .firstChild property and change the .textContent property of the first child. |

|  |  |
| --- | --- |
| Minimax Notes Simulates every possible move and sub move and in the case that such a move would result in a win or loss, a value is returned corresponding to that win or loss.   1. Make a move 2. If that move results in a win or loss or full board return a value (-,0,+) 3. If there is no result from that move, simulate the above actions for every possible sub-move and store the value of the sub move with the highest value   The reason why the smaller value is being stored when the sub-moves are being simulated for the enemy is because the enemy wants to make the AI lose and a loss results in a negative value.  When the next move in the simulation will be made by the enemy, store the smallest value as the enemies goal is to minimize your/AI score.   |  | | --- | | bestMoveValue = Math.min(minimax(x,y,enemyTurn=true,….); | |

Even when declaring objects with the notation of var obj = {}, any functions inside that object must reference object member variables using the .this keyword.

Strange issue of console.logging an array modified or used somewhere else. The array seems to retain the value it had at a certain place regardless of any changes done to it.

Bellow, the first console.log() displays the same value as the one after it even though it had not been modified yet?? WHY??

|  |
| --- |
| **var arr = [["X","X","-"], ["-","-","O"], ["X","X","O"] ];**  **console.log(arr);**  **ModIt(arr);**  **function ModIt(pureBoard)**  **{**  **pureBoard[0][2] = "X";**  **console.log(pureBoard);**  **}** |

End game animation won’t play on Internet Explorer. I don’t really care but adding a warning would be nice.

Tic-Tac-Toe white inner boarders of the Play Screen will overlap the header unless the header’s z-position is set to a higher number. This only occurs with the inner white boarders and no other boarders or elements?