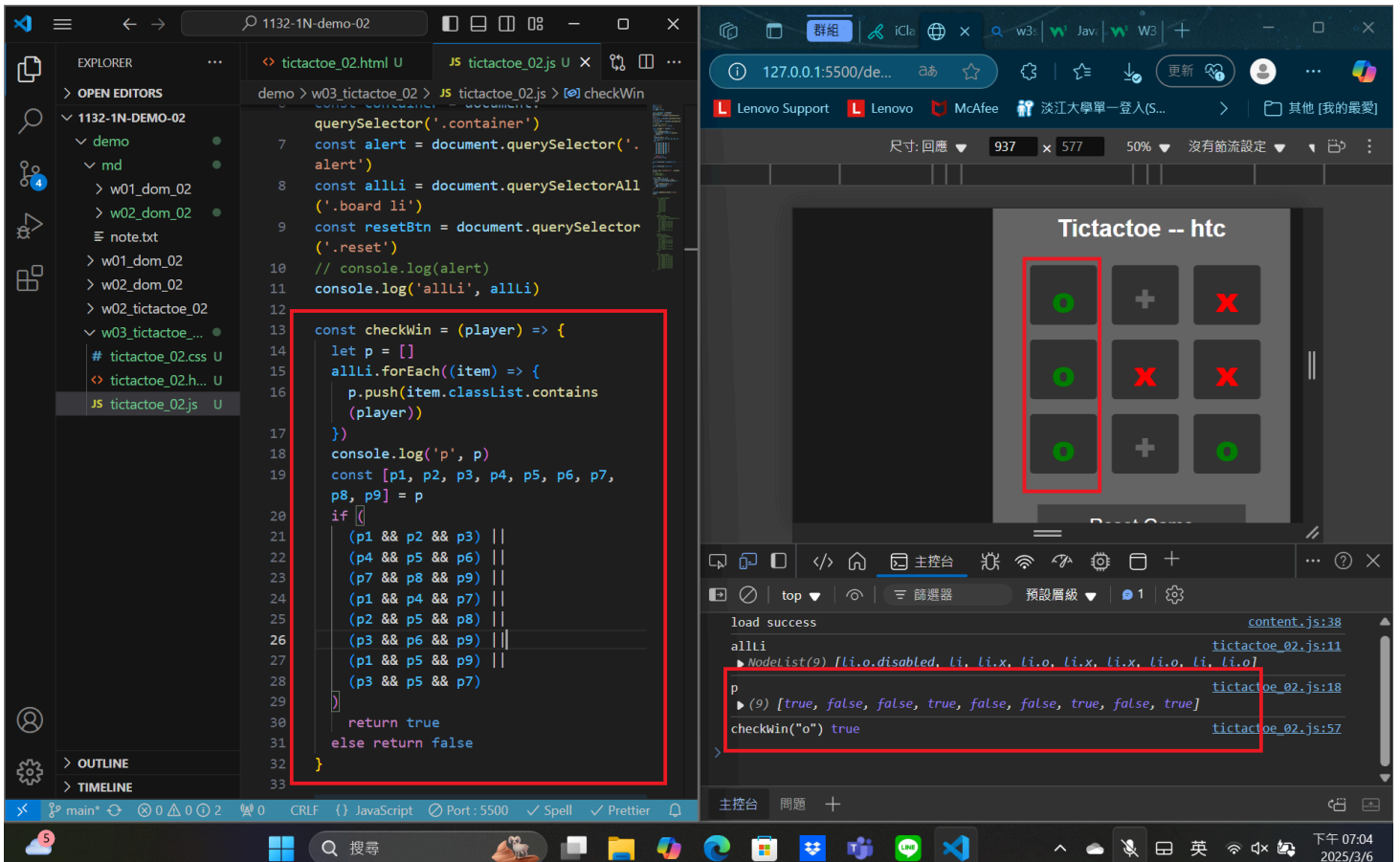


Video: W03-P1: Implement checkWin(player) using three different cases

=> player o wins



=> player x wins

The image shows a development environment with VS Code on the left and a web browser on the right. The VS Code editor displays the file explorer on the left with the project structure: 1132-1N-DEMO-02, demo, md, w01_dom_02, w02_dom_02, note.txt, w03-p1-1.png, w01_dom_02, w02_dom_02, w02_tictactoe_02, w03_tictactoe_02, # tictactoe_02.css, < tictactoe_02.h..., JS tictactoe_02.js. The main editor shows the code for tictactoe_02.js, with the checkWin function and its calls highlighted. The browser shows the game interface with a 3x3 grid and a console log showing the game state.

```
demo > w03_tictactoe_02 > JS tictactoe_02.js > [0] reset
13 const checkWin = (player) => {
14   let p = []
15   allLi.forEach((item) => {
16     p.push(item.classList.contains
17       (player))
18   })
19   console.log('p', p)
20   const [p1, p2, p3, p4, p5, p6, p7,
21     p8, p9] = p
22   if (
23     (p1 && p2 && p3) ||
24     (p4 && p5 && p6) ||
25     (p7 && p8 && p9) ||
26     (p1 && p4 && p7) ||
27     (p2 && p5 && p8) ||
28     (p3 && p6 && p9) ||
29     (p1 && p5 && p9) ||
30     (p3 && p5 && p7)
31   )
32     return true
33   else return false
34 }
35
36 > const winMessage = (player) => { ...
48 }
49
50 > const tieMessage = () => { ...
55 }
56
57 console.log('checkWin("o")', checkWin
58   ('o'))
59 console.log('checkWin("x")', checkWin
60   ('x'))
61 // tieMessage()
```

The browser shows the game interface with a 3x3 grid. The grid contains the following symbols: Row 1: O, +, X; Row 2: O, X, +; Row 3: X, +, O. The console log shows the game state: (9) [true, false, false, true, false, false, false, false, true], checkWin("o") false, (9) [false, false, true, false, true, false, true, false, false], checkWin("x") true.

=> no player wins

The screenshot displays a web development environment with a code editor on the left and a browser on the right. The code editor shows the implementation of the `checkWin` function in `tictactoe_02.js`. The function checks for a win condition by iterating over the board state `p` and checking for three 'O's or three 'X's in a row, column, or diagonal. The browser shows the game board with 'O' and 'X' pieces. The console shows the results of the `checkWin` function calls.

```
const checkWin = (player) => {
  let p = []
  allli.forEach((item) => {
    p.push(item.classList.contains(player))
  })
  console.log('p', p)
  const [p1, p2, p3, p4, p5, p6, p7, p8, p9] = p
  if ([
    (p1 && p2 && p3) ||
    (p4 && p5 && p6) ||
    (p7 && p8 && p9) ||
    (p1 && p4 && p7) ||
    (p2 && p5 && p8) ||
    (p3 && p6 && p9) ||
    (p1 && p5 && p9) ||
    (p3 && p5 && p7)
  ]) {
    return true
  } else {
    return false
  }
}

const winMessage = (player) => { ... }
const tieMessage = () => { ... }

console.log('checkWin("o")', checkWin('o'))
console.log('checkWin("x")', checkWin('x'))
```

The browser shows the game board with 'O' and 'X' pieces. The console shows the results of the `checkWin` function calls:

```
p ▶ (9) [true, false, true, true, true, false, false, true, false] tictactoe_02.js:18
checkWin("o") false tictactoe_02.js:57
p ▶ (9) [false, true, false, false, false, true, true, false, true] tictactoe_02.js:18
checkWin("x") false tictactoe_02.js:58
```

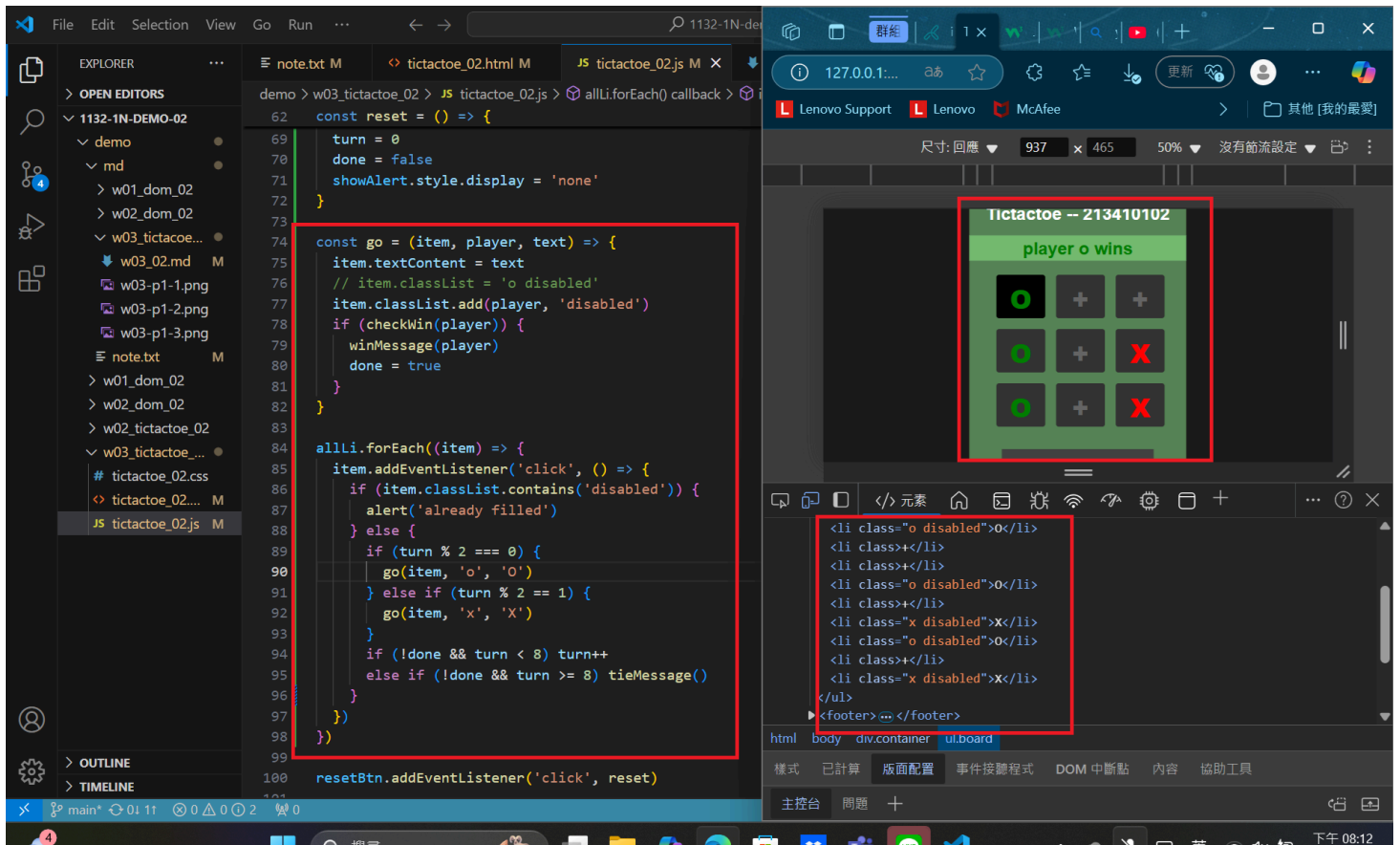
0a5bae1 apple550678

Thu Mar 6 19:15:06 2025 +0800

Video: W03-P1: Implement checkWin(player

Video: W03-P2: play TicTacToe successfully

=> player o wins



=> player x wins

The image displays a web development environment with two main panes. The left pane shows the file explorer and the code editor. The right pane shows the browser preview of the game interface.

Code Editor (Left Pane):

```
demo > w03_tictactoe_02 > JS tictactoe_02.js > allLi.forEach() callback >
62 const reset = () => {
71   showAlert.style.display = 'none'
72 }
73
74 const go = (item, player, text) => {
75   item.textContent = text
76   // item.classList = 'o disabled'
77   item.classList.add(player, 'disabled')
78   if (checkWin(player)) {
79     winMessage(player)
80     done = true
81   }
82 }
83
84 allLi.forEach((item) => {
85   item.addEventListener('click', () => {
86     if (item.classList.contains('disabled')) {
87       alert('already filled')
88     } else {
89       if (turn % 2 === 0) {
90         go(item, 'o', 'O')
91       } else if (turn % 2 === 1) {
92         go(item, 'x', 'X')
93       }
94       if (!done && turn < 8) turn++
95       else if (!done && turn >= 8) tieMessage()
96     }
97   })
98 })
99
100 resetBtn.addEventListener('click', reset)
```

Browser Preview (Right Pane):

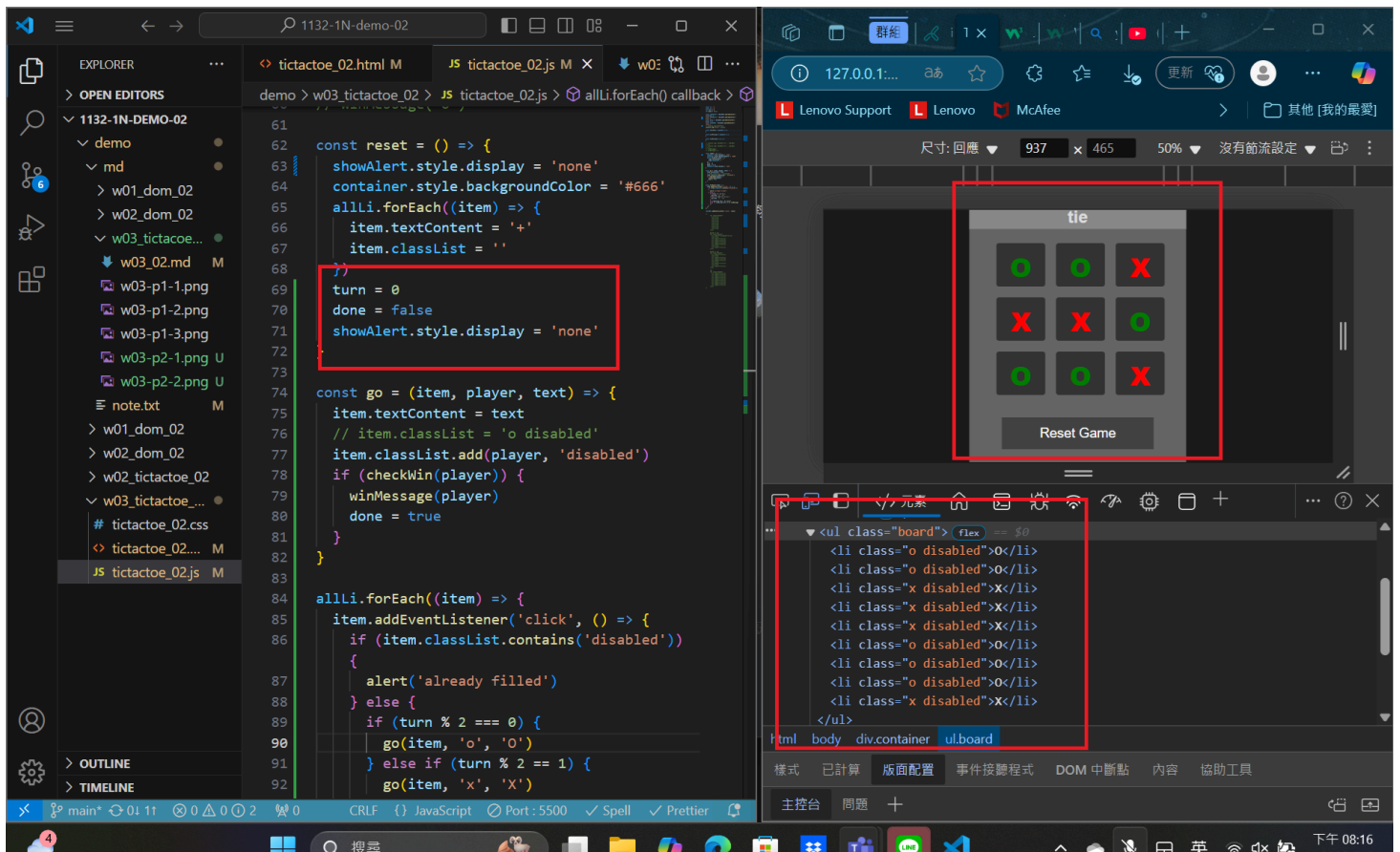
The browser shows a game interface titled "Tictactoe -- 213410102". The game board displays "player x wins". The board is a 3x3 grid with the following state:

O	+	X
O	+	X
+	O	X

The DOM inspector shows the following HTML structure for the board:

```
<ul class="board">
  <li class="o disabled">O</li>
  <li>+</li>
  <li class="x disabled">X</li>
  <li class="o disabled">O</li>
  <li>+</li>
  <li class="x disabled">X</li>
  <li>+</li>
  <li class="o disabled">O</li>
  <li class="x disabled">X</li>
</ul>
```

=> tie



55965a2 apple550678

Thu Mar 6 20:18:05 2025 +0800

Video: W03-P2: play TicTacToe successful

Video: W03-P3: Temperature Convert from C to F

