

W02-P1: Show Tictactoe css

```

# tictactoe_31.css
body {
    background: #f1f1f1;
    color: #666;
    font-family: arial;
}

header h1 {
    text-align: center;
    color: #fff;
    padding-top: 20px;
}

.container {
    width: 400px;
    background-color: #666;
    margin: 40px auto; /*auto置中*/
    border-radius: 10px;
    padding-bottom: 40px;
}

.board {
    display: flex;
    flex-wrap: wrap;
}

.board li {
    height: 70px;
    width: 90px;
    margin: 10px;
    background-color: #333;
    text-align: center; /*文字置中*/
    font-size: 48px;
    font-weight: 800px;
    list-style: none;
    padding-top: 20px;
    border-radius: 5px;
}

.footer {
}

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="http://127.0.0.1:5500/demo/w02-tictactoe_31/tictactoe_31.css" cacheOverride=1709203974511">
</head>
<body>
    <div id="container">
        <header>
            <h1>Tictactoe Liang Y</h1>
        </header>
        <ul id="board">flex
            <li class="o">O</li>
            <li>+</li>
            <li>+</li>
            <li>+</li>
            <li>+</li>
            <li>+</li>
            <li>+</li>
            <li>+</li>
        </ul>
        <button id="reset">Rest Game</button>
        <script src="/tictactoe_31.js"></script>
    </div>
    <!-- Code injected by live-server -->
    <script>...</script> == $0
</body>
</html>

```

4d4c174 陳亮瑜 Thu Feb 29 19:02:34 2024 +0800 ### W02-P1: Show Tictactoe css

W02-P2: implement reset button

The screenshot shows a web browser window with a Tic-Tac-Toe game titled "Tictactoe Liang Yu". The game board is a 3x3 grid of buttons, each containing a plus sign ("+"). A red box highlights the entire game board area. Below the board is a "Rest Game" button.

On the right side of the browser, there is a code editor window titled "tictactoe_31.html M" and "tictactoe_31.js M". The HTML code defines the game board structure:

```
<div id="container" style="background-color: #ccc; width: 300px; height: 300px; margin: auto; border: 1px solid black; padding: 10px; border-radius: 10px; display: flex; flex-wrap: wrap; align-items: center; justify-content: center; gap: 10px; font-size: 2em; font-weight: bold; font-family: sans-serif; text-align: center; margin-top: 20px;>
```

The JavaScript code contains logic for the game, including a reset function:

```
const reset = () => {
    //點reset, class會變空值
    allLi[0].classList = '';
    當reset會變回+號，改變文字內容
    allLi[0].textContent = '+';
    allLi[4].classList = '';
    allLi[4].textContent = '+';
}

/*item每一個值[0]~[8]*/
allLi.forEach((item) => {
    item.classList = '';
    item.textContent = '+';
});

//css搬過來要去掉“-”符號,第二個字開頭變大寫
container.style.backgroundColor = '#666';
turn = 0;
done = false;
};

//當'click'後,交給reset function
resetBtn.addEventListener('click', reset);
```

5ea6097 陳亮瑜 Thu Feb 29 19:59:15 2024 +0800 ### W02-P2: implement reset button

W02-P3: implement checkWin(player), and winMessage(player)

=> player O wins

The screenshot shows a code editor and a browser window side-by-side.

Code Editor (left):

```
const checkWin = (player) => {
  /*p是一個陣列*/
  let p = [];
  allLi.forEach(item => {
    p.push(item.classList.contains(player));
  });
  /*抓到值是x還是o,true or false*/
  console.log('p', p);
  /*@用p1取代,1用o2取代....*/
  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到checkWin */
    return true;
  else return false;
};

const winMessage = (player) => {
  if (player === 'o') {
    container.style.backgroundColor = 'rgba(144, 238, 144, 0.5)';
  } else {
    container.style.backgroundColor = 'rgba(240, 118, 128, 0.726)';
  }
  alert(`player ${player} wins`);
};

allLi.forEach((item) => {
  /*監控每一個item*/
  item.addEventListener('click', () => {
    if (item.classList.contains('disabled')) {
      alert('already filled');
    } else {
      if (turn % 2 === 0) {
        /*偶數, === 指定*/
        item.textContent = 'O';
      } else {
        item.textContent = 'X';
      }
    }
  });
};
```

Browser Preview (right):

The browser shows a Tic-Tac-Toe game titled "Tictactoe Liang Yu". The board has three rows and three columns of buttons. The first row contains 'O', 'X', and '+'. The second row contains 'O', '+', 'X'. The third row contains 'O', '+', '+'. A "Rest Game" button is at the bottom. The developer tools' element panel highlights the board area, and the console panel shows the state of the 'p' array:

```
p [9] [true, false, false, false, false, false, false, false, false]
  0: true
  1: false
  2: false
  3: true
  4: false
  5: false
  6: true
  7: false
  8: false
length: 9
```

=> player X wins

The screenshot shows a browser window displaying a Tic-tac-toe game. The game board consists of a 3x3 grid of buttons. The top row contains two 'O' symbols and one '+' symbol. The middle row contains three 'X' symbols. The bottom row contains one '+' symbol, one 'O' symbol, and one '+' symbol. A red box highlights the middle row of buttons. Below the board is a 'Rest Game' button.

Left panel (Code Editor):

```
1.css w02_31.md M tictactoe_31.html M tictactoe_31.js M
12-tictactoe_31 > JS tictactoe_31.js > allLi.forEach() callback > item.addEventListener('click') callback
41 const checkWin = (player) => {
42     /*p是一個陣列*/
43     let p = [];
44     allLi.forEach((item) => {
45         p.push(item.classList.contains(player));
46     });
47     /*抓到值是x還是o,true or false*/
48     console.log('p', p);
49     /*@用p1取代,1用p2取代....*/
50     const [p1, p2, p3, p4, p5, p6, p7, p8, p9] = p;
51     if (
52         (p1 && p2 && p3) ||
53         (p4 && p5 && p6) ||
54         (p7 && p8 && p9) ||
55         (p1 && p4 && p7) ||
56         (p2 && p5 && p8) ||
57         (p3 && p6 && p9) ||
58         (p1 && p5 && p9) ||
59         (p3 && p5 && p7)
60     )
61     /*return true@checkWin */
62     return true;
63     else return false;
64 };
65
66 const winMessage = (player) => {
67     if (player === 'o') {
68         container.style.backgroundColor = 'rgba(144, 238, 144, 0.5)';
69     } else {
70         container.style.backgroundColor = 'rgba(240, 118, 128, 0.726)';
71     }
72     alert(`player ${player} wins`);
73 };
74
75 allLi.forEach((item) => {
76     /*監控每一個item*/
77     item.addEventListener('click', () => {
78         if (item.classList.contains('disabled')) {
79             alert('already filled');
80         } else {
81             if (turn % 2 === 0) {
82                 /*偶數,=""指定*/
83                 item.textContent = 'o';
84             } else {
85                 item.textContent = 'x';
86             }
87         }
88     });
89 });
90
91 turn++;
92
93 if (turn === 9) {
94     alert('Game Over');
95 }
```

Right panel (Browser DevTools):

```
Document 127.0.0.1:5500/demo/w02-tictactoe_31/tictactoe_31.html
Container tictactoe_31.js:16
  <div id="container"> ...
  allLi tictactoe_31.js:17
    > NodeList(9) [li, li, li, li, li, li, li, li, li]
  restBtn tictactoe_31.js:18
    <button id="reset">Rest Game
    </button>
  p tictactoe_31.js:48
    (9) [true, false, false, false, false, false, false, false, false]
  p tictactoe_31.js:48
    (9) [false, false, false, false, false, false, false, false, false]
  p tictactoe_31.js:48
    (9) [true, true, false, false, false, false, false, false, false]
  p tictactoe_31.js:48
    (9) [false, false, false, false, false, false, false, false, false]
  p tictactoe_31.js:48
    (9) [true, true, false, false, false, false, false, false, false]
  p tictactoe_31.js:48
    (9) [false, false, false, false, false, false, false, false, false]
  p tictactoe_31.js:48
    (9) [false, false, false, true, true, true, true, true, true]
    0: false
    1: false
    2: false
    3: true
    4: true
    5: true
    6: false
    7: false
    8: false
    length: 9
    > [[Prototype]: Array(0)]
```

=> tie game

The screenshot shows a browser window displaying a Tictactoe game. The game board is a 3x3 grid of buttons. The top row contains 'X', 'O', 'O'. The middle row contains 'O', 'X', 'X'. The bottom row contains 'X', 'O', '+'. A red box highlights the entire grid. Above the grid, a message box says '127.0.0.1:5500 顯示 tie' (127.0.0.1:5500 displays tie). To the right of the browser, the developer tools are open, showing the JavaScript code for the game. A specific section of the code is highlighted with a red box:

```
    } else if (!done && turn >= 8) {
        /*若走完還沒結束就是tie*/
        alert('tie');
    }
});
```

The browser's status bar at the bottom shows: main* 0 0 4 0 0 LF () JavaScript Port: 5500 Spell Prettier

443eefe 陳亮瑜 Thu Feb 29 21:01:03 2024 +0800 ### W02-P3: implment checkWin(player), a

W02-Log:

git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-02-27"

443eefe 陳亮瑜 Thu Feb 29 21:01:03 2024 +0800 ### W02-P3: implment checkWin(player), a
5ea6097 陳亮瑜 Thu Feb 29 19:59:15 2024 +0800 ### W02-P2: implement reset button
4d4c174 陳亮瑜 Thu Feb 29 19:02:34 2024 +0800 ### W02-P1: Show Tictactoe css