# Week 11: Coding Assignment

**URL to GitHub Repository: https://github.com/SpaceChkn/week11**

**URL to Your Coding Assignment Video:**
**https://www.dropbox.com/s/e1skhnso6tk5a4w/week11.mp4?dl=0**

**Instructions:**

- In Visual Studio Code, write the code that accomplishes the objectives listed below and ensures that the code compiles and runs as directed.
- Create a new repository on GitHub for this week's assignments and push this document, with your project code, to the repository.
- Include the URLs for this week's repository and video where instructed.
- Submit this document as a .PDF file in the LMS.

**Coding Steps:**

- Using any of the tools you've worked with so far, create a game of Tic-Tac-Toe.
    - Create a Tic-Tac-Toe game grid using your HTML element of choice.
    - When a cell in the grid is clicked, an **X** or **O** should appear in that spot depending on whose turn it is.
    - A heading should say whether it is X's or O's turn and change with each move made.
    - A button should be available to clear the grid and restart the game.
    - When a player has won, or the board is full and the game results in a draw, a Bootstrap alert or similar Bootstrap component should appear across the screen announcing the winner.

# Week 11: Coding Assignment

**Video Steps:**

- Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed.
- This video should be done using screen share and voice over.
- This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend.
    - You can create a new meeting, start screen sharing, and start recording.
    - This will create a video recording on your computer.
- This should then be uploaded to a publicly accessible site, such as YouTube.
    - Ensure the link you share is **PUBLIC** or **UNLISTED**!
    - If it is not accessible by your grader, your project will be graded based on what they can access.

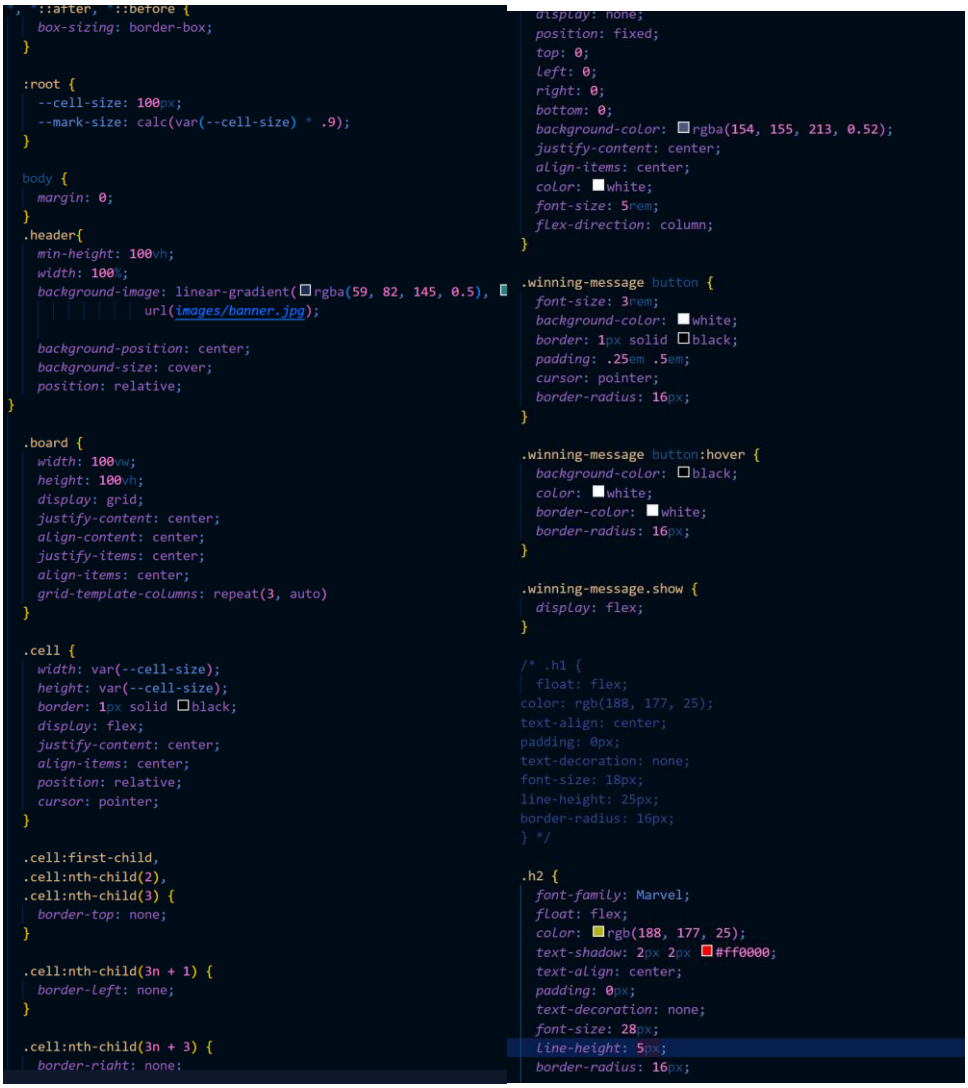
PROMINEO TECH

# Week 11: Coding Assignment

```
const CIRCLE_CLASS = 'circle'
const WINNING_COMBINATIONS = [
  [0, 1, 2],
  [3, 4, 5],
  [6, 7, 8],
  [0, 3, 6],
  [1, 4, 7],
  [2, 5, 8],
  [0, 4, 8],
  [2, 4, 6]
]
const cellElements = document.querySelectorAll('[data-cell]')
const board = document.getElementById('board')
const winningMessageElement = document.getElementById('winningMessage')
const restartButton = document.getElementById('restartButton')
const winningMessageTextElement = document.querySelector('[data-winning-message-text]')
let circleTurn

startGame()

restartButton.addEventListener('click', startGame)

function startGame() {
  circleTurn = false
  cellElements.forEach(cell => {
    cell.classList.remove(X_CLASS)
    cell.classList.remove(CIRCLE_CLASS)
    cell.removeEventListener('click', handleClick)
    cell.addEventListener('click', handleClick, { once: true })
  })
  setBoardHoverClass()
  winningMessageElement.classList.remove('show')
}

function handleClick(e) {
  const cell = e.target
  const currentClass = circleTurn ? CIRCLE_CLASS : X_CLASS
  placeMark(cell, currentClass)
  if (checkWin(currentClass)) {
    endGame(false)
  } else if (isDraw()) {
    endGame(true)
  } else {
    swapTurns()
    setBoardHoverClass()
  }
}

function endGame(draw) {
  if (draw) {
    winningMessageTextElement.innerText = 'Draw!'
  } else {
    winningMessageTextElement.innerText = `${circleTurn ? "O" : "X"} Wins!`
  }
  winningMessageElement.classList.add('show')
}
```

```
  } else {
    winningMessageTextElement.innerText = `${circleTurn ? "O" : "X"} Wins!`
  }
  winningMessageElement.classList.add('show')
}

function isDraw() {
  return [...cellElements].every(cell => {
    return cell.classList.contains(X_CLASS) || cell.classList.contains(CIRCLE_CLASS)
  })
}

function placeMark(cell, currentClass) {
  cell.classList.add(currentClass);
  if (currentClass == "x") {
    document.getElementById('Player').innerHTML = "Player O";
  } else {
    if (currentClass == "circle") {document.getElementById('Player').innerHTML = "Player X
    }

  }
    // console.log(currentClass);
  }



function swapTurns() {
  circleTurn = !circleTurn
}

// function Turns() {
//     isPlayerOTurn = !isPlayerOTurn
//     document.getElementById('turn').innerHTML = `player ${isPlayerOTurn ? "O": "X"}`
// }


function setBoardHoverClass() {
  board.classList.remove(X_CLASS)
  board.classList.remove(CIRCLE_CLASS)
  if (circleTurn) {
    board.classList.add(CIRCLE_CLASS)
  } else {
    board.classList.add(X_CLASS)
  }
}

function checkWin(currentClass) {
  return WINNING_COMBINATIONS.some(combination => {
    return combination.every(index => {
      return cellElements[index].classList.contains(currentClass)
    })
  })
}
```

# Week 11: Coding Assignment



```
*, ::after, ::before {
  box-sizing: border-box;
}

:root {
  --cell-size: 100px;
  --mark-size: calc(var(--cell-size) * .9);
}

body {
  margin: 0;
}
.header{
  min-height: 100vh;
  width: 100%;
  background-image: linear-gradient(rgba(59, 82, 145, 0.5),
                    url(images/banner.jpg);

  background-position: center;
  background-size: cover;
  position: relative;
}

.board {
  width: 100vw;
  height: 100vh;
  display: grid;
  justify-content: center;
  align-content: center;
  justify-items: center;
  align-items: center;
  grid-template-columns: repeat(3, auto)
}

.cell {
  width: var(--cell-size);
  height: var(--cell-size);
  border: 1px solid black;
  display: flex;
  justify-content: center;
  align-items: center;
  position: relative;
  cursor: pointer;
}

.cell:first-child,
.cell:nth-child(2),
.cell:nth-child(3) {
  border-top: none;
}

.cell:nth-child(3n + 1) {
  border-left: none;
}

.cell:nth-child(3n + 3) {
  border-right: none;
```

```
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: rgba(154, 155, 213, 0.52);
  justify-content: center;
  align-items: center;
  color: white;
  font-size: 5rem;
  flex-direction: column;
}

.winning-message button {
  font-size: 3rem;
  background-color: white;
  border: 1px solid black;
  padding: .25em .5em;
  cursor: pointer;
  border-radius: 16px;
}

.winning-message button:hover {
  background-color: black;
  color: white;
  border-color: white;
  border-radius: 16px;
}

.winning-message.show {
  display: flex;
}

/* .h1 {
  float: flex;
color: rgb(188, 177, 25);
text-align: center;
padding: 0px;
text-decoration: none;
font-size: 18px;
line-height: 25px;
border-radius: 16px;
} */

.h2 {
  font-family: Marvel;
  float: flex;
  color: rgb(188, 177, 25);
  text-shadow: 2px 2px #ff0000;
  text-align: center;
  padding: 0px;
  text-decoration: none;
  font-size: 28px;
  line-height: 5px;
  border-radius: 16px;
```

# Week 11: Coding Assignment