GAME GARDEN

CHIOREAN-PÉTER NOÉMI
CSISZÉR BÁLINT
JÁNOSI-BARNA BOTOND
SZÖVÉRFI SAROLTA

WHAT KNOWLEDGE WAS USED TO CREATE THE PAGE?

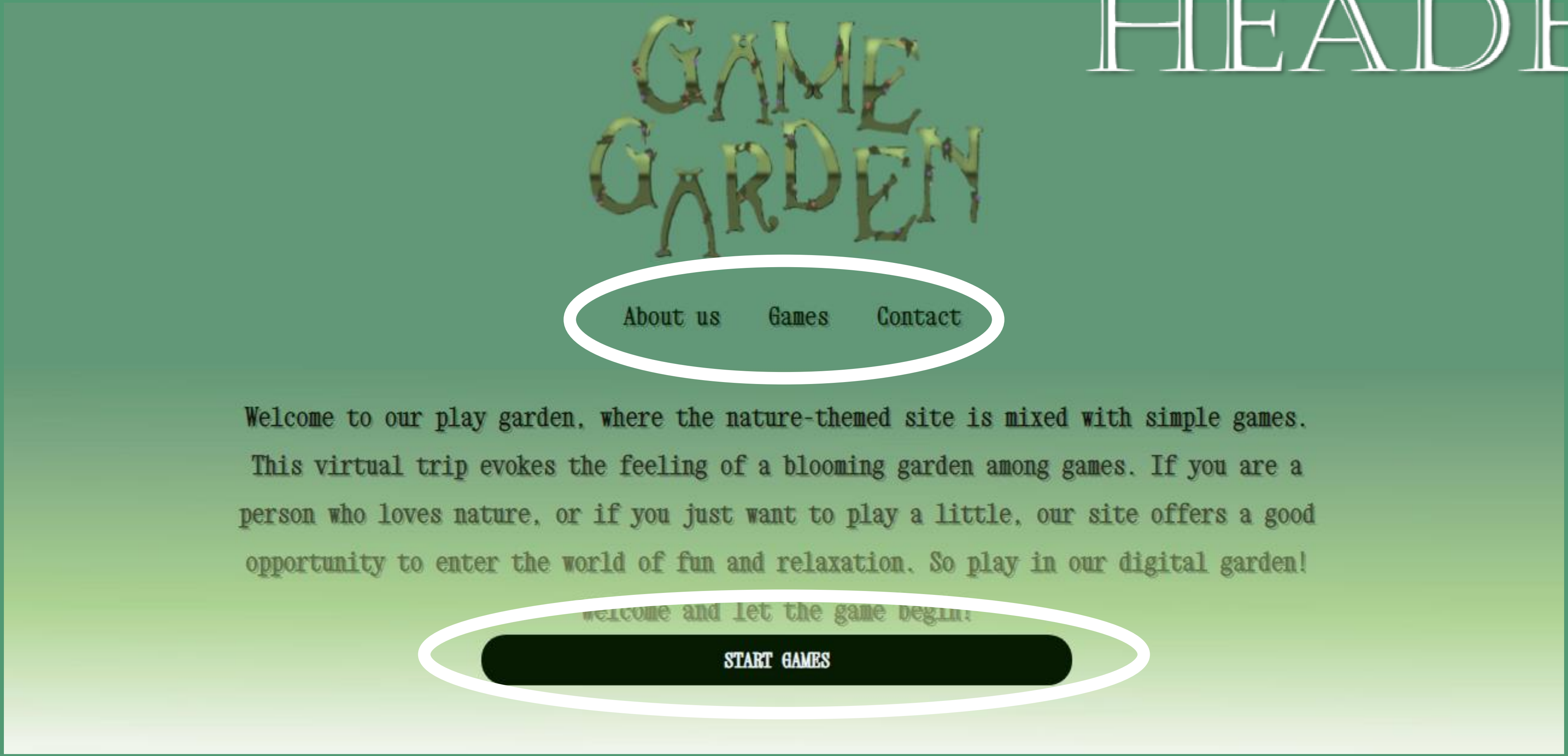
1. knowledge learned in school
2. knowledge learned here
3. knowledge learned based on tutorials

} CSS,HTML, JS

WHAT IS THE PURPOSE OF THE SITE?

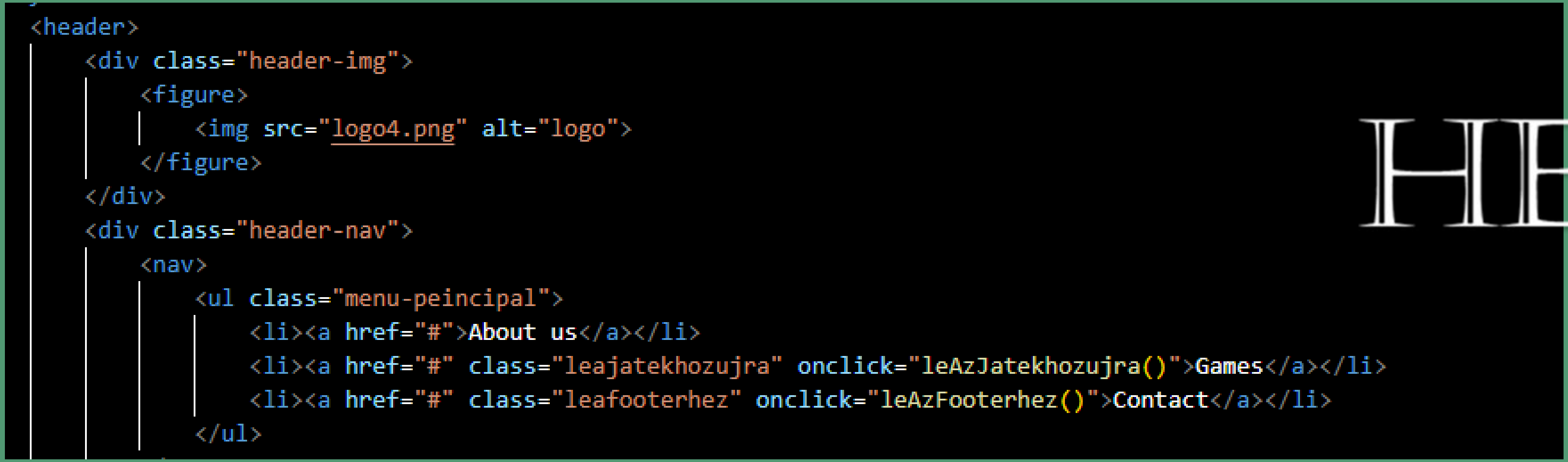
Using the basics of JavaScript, HTML and CSS to create a website with games that uses what we have learned here.

WEBSITE STRUCTURE: HEADER

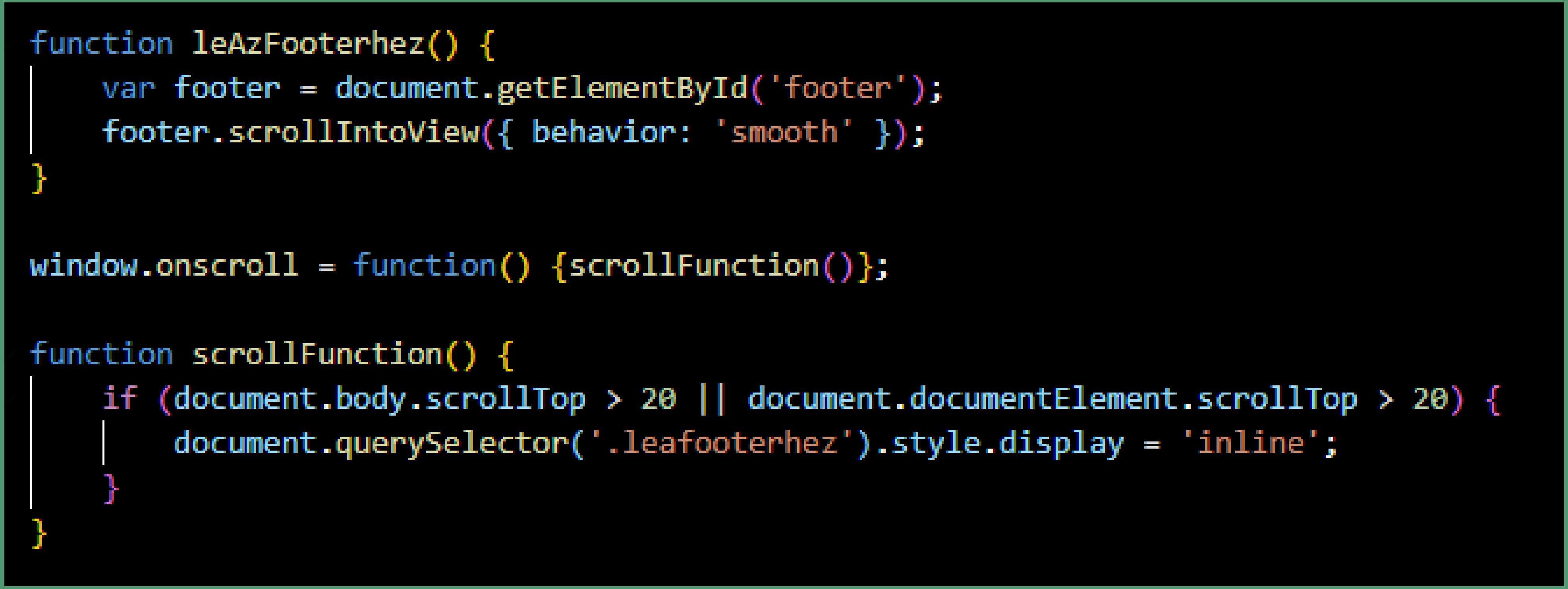


Welcome to our play garden, where the nature-themed site is mixed with simple games. This virtual trip evokes the feeling of a blooming garden among games. If you are a person who loves nature, or if you just want to play a little, our site offers a good opportunity to enter the world of fun and relaxation. So play in our digital garden! Welcome and let the game begin!

HEADER

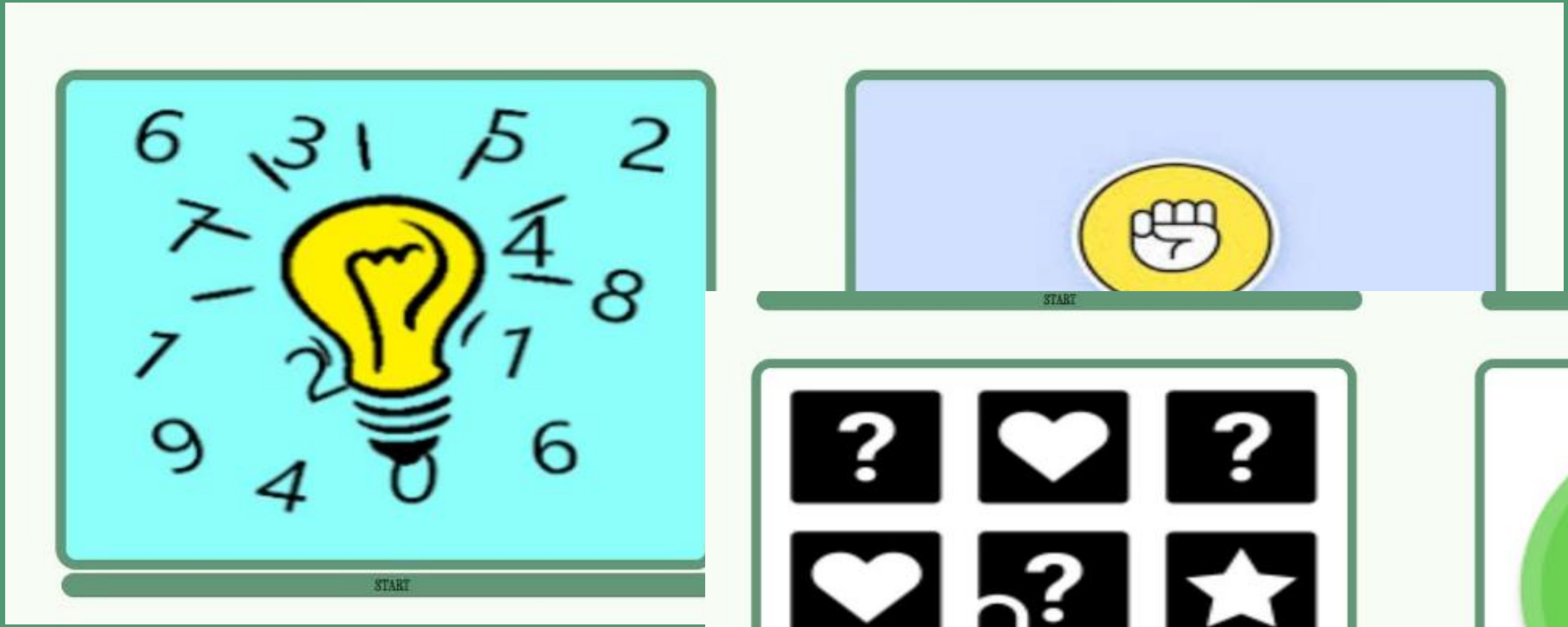


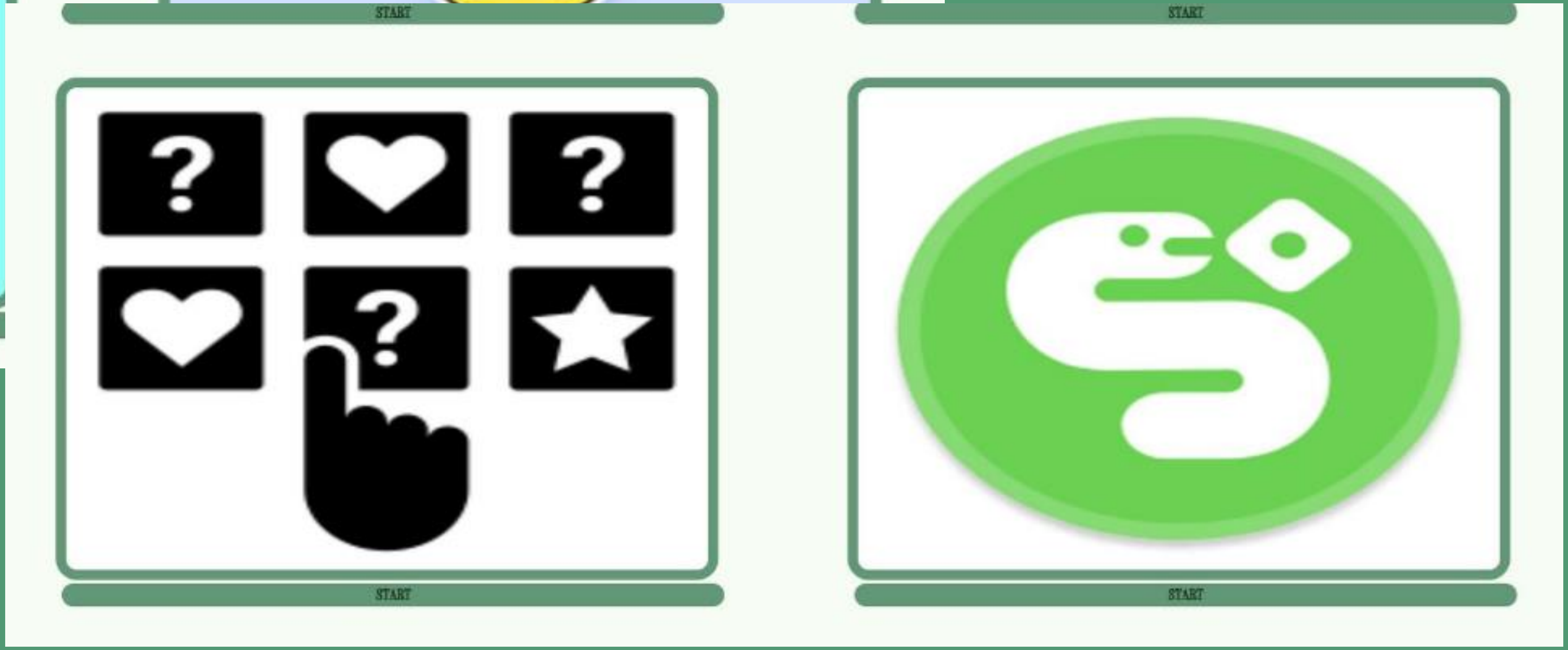
```html
<header>
    <div class="header-img">
        <figure>
            <img src="logo4.png" alt="logo">
        </figure>
    </div>
    <div class="header-nav">
        <nav>
            <ul class="menu-peincipal">
                <li><a href="#">About us</a></li>
                <li><a href="#" class="leajatekhozujra" onclick="leAzJatekhozujra()">Games</a></li>
                <li><a href="#" class="leafooterhez" onclick="leAzFooterhez()">Contact</a></li>
            </ul>
```

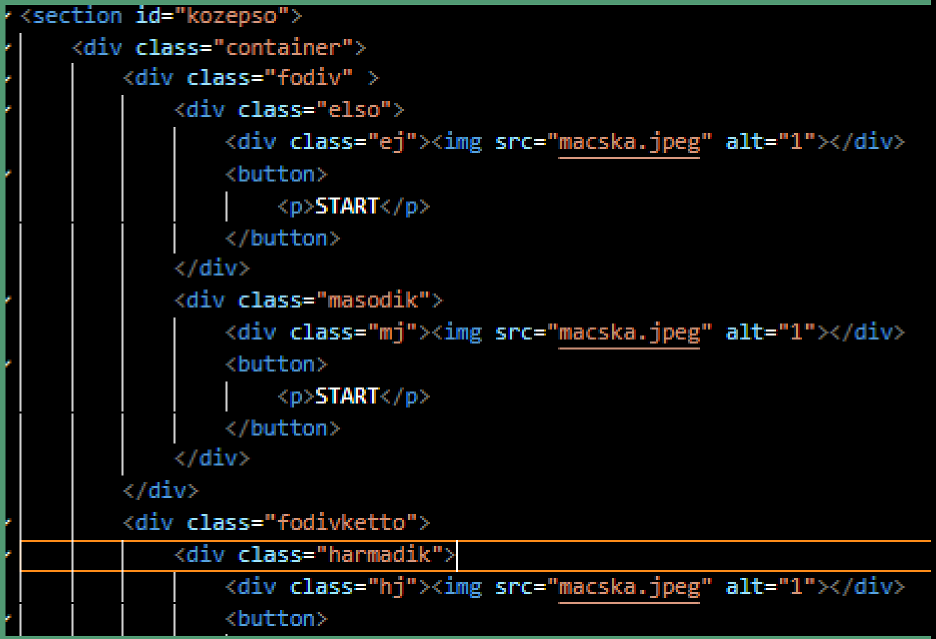


```javascript
function leAzFooterhez() {
    var footer = document.getElementById('footer');
    footer.scrollIntoView({ behavior: 'smooth' });
}

window.onscroll = function() {scrollFunction()};

function scrollFunction() {
    if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
        document.querySelector('.leafooterhez').style.display = 'inline';
    }
}
```

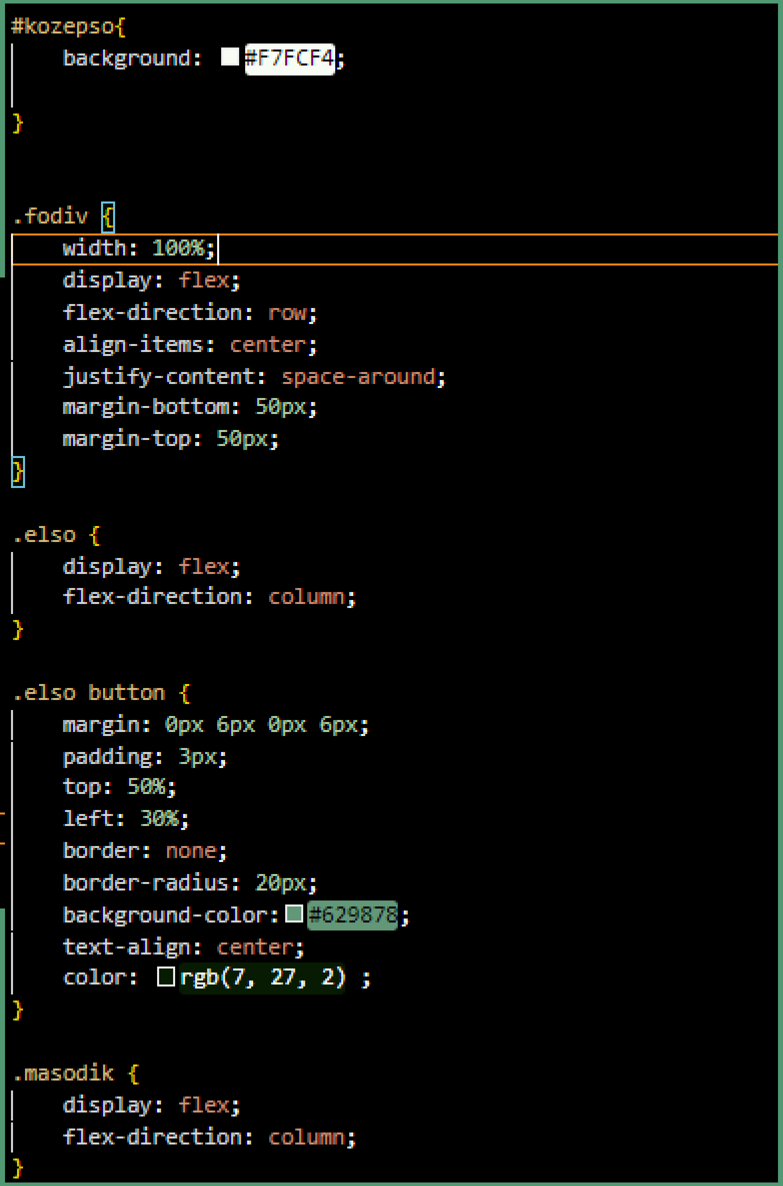
MAIN PART OF THE WEBPAGE



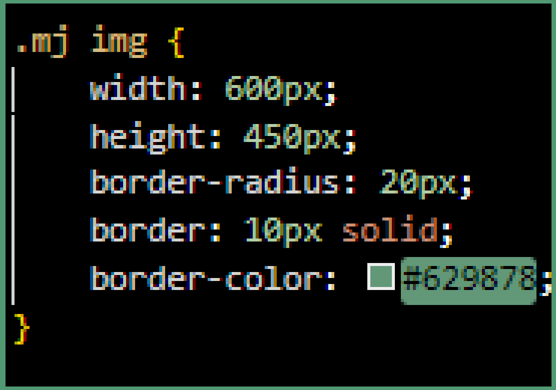




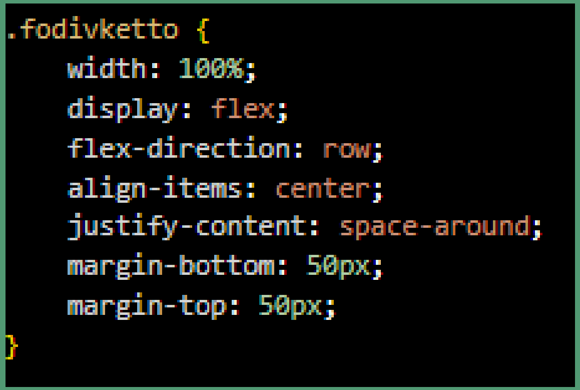
```html
<section id="kozepso">
    <div class="container">
        <div class="fodiv" >
            <div class="elso">
                <div class="ej"><img src="macska.jpeg" alt="1"></div>
                <button>
                    <p>START</p>
                </button>
            </div>
            <div class="masodik">
                <div class="mj"><img src="macska.jpeg" alt="1"></div>
                <button>
                    <p>START</p>
                </button>
            </div>
        </div>
        <div class="fodivketto">
            <div class="harmadik">
                <div class="hj"><img src="macska.jpeg" alt="1"></div>
                <button>
```



```css
#kozepso{
    background: #F7FCF4;

}


.fodiv {
    width: 100%;
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: space-around;
    margin-bottom: 50px;
    margin-top: 50px;
}

.elso {
    display: flex;
    flex-direction: column;
}

.elso button {
    margin: 0px 6px 0px 6px;
    padding: 3px;
    top: 50%;
    left: 30%;
    border: none;
    border-radius: 20px;
    background-color:#629878;
    text-align: center;
    color: rgb(7, 27, 2) ;
}

.masodik {
    display: flex;
    flex-direction: column;
}
```

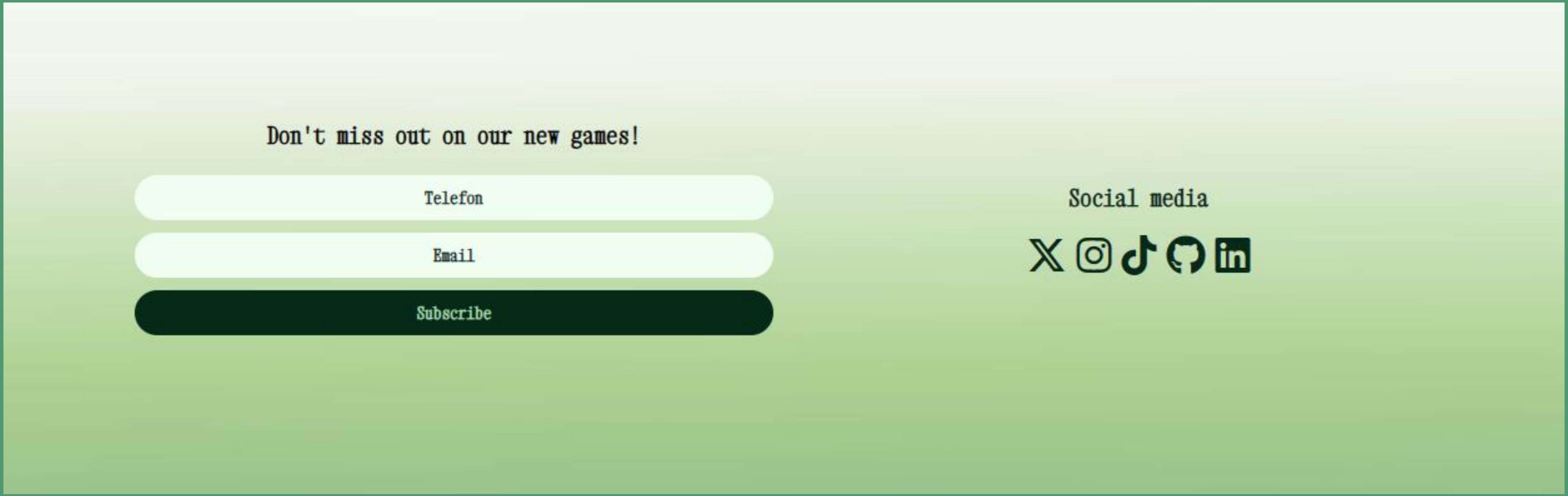


```css
.mj img {
    width: 600px;
    height: 450px;
    border-radius: 20px;
    border: 10px solid;
    border-color: #629878;
}
```



```css
.fodivketto {
    width: 100%;
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: space-around;
    margin-bottom: 50px;
    margin-top: 50px;
}
```

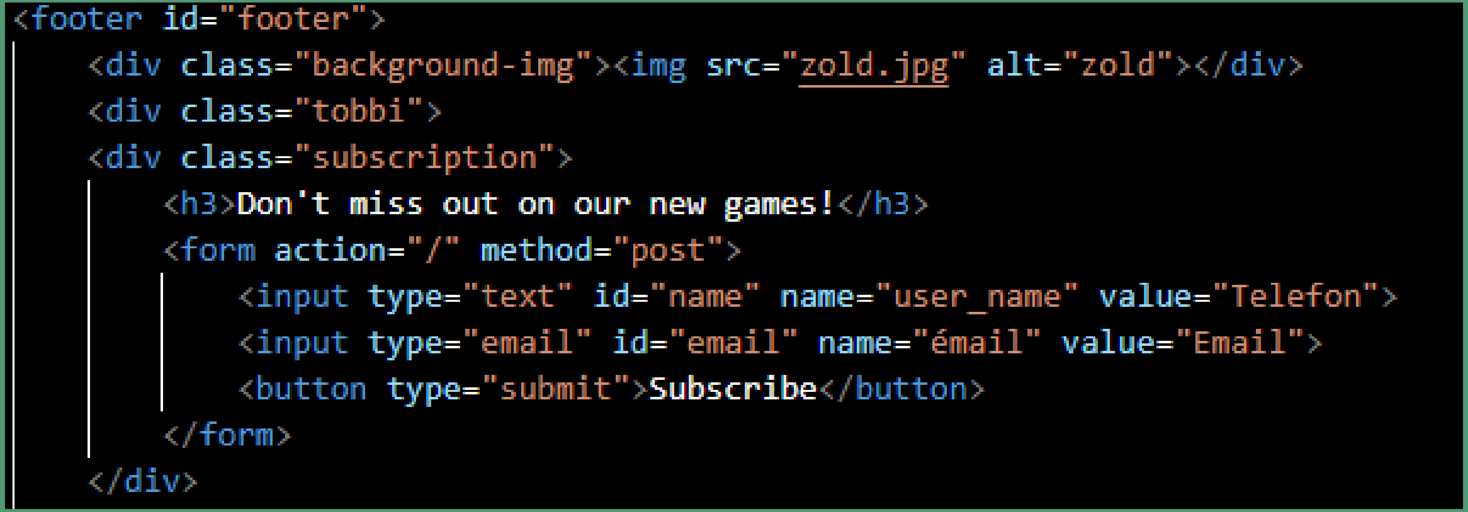
FOOTER



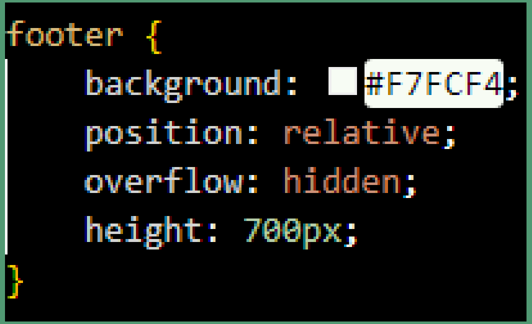
Don't miss out on our new games!

Social media

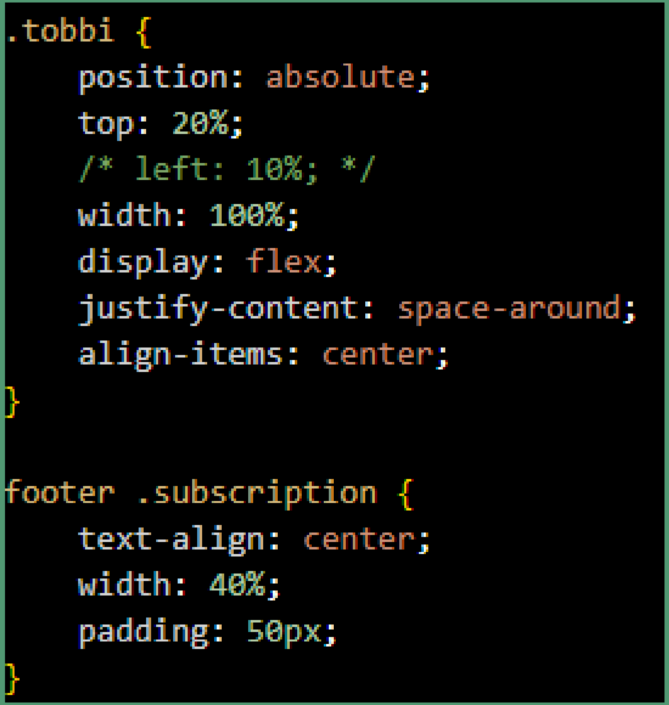
FOOTER



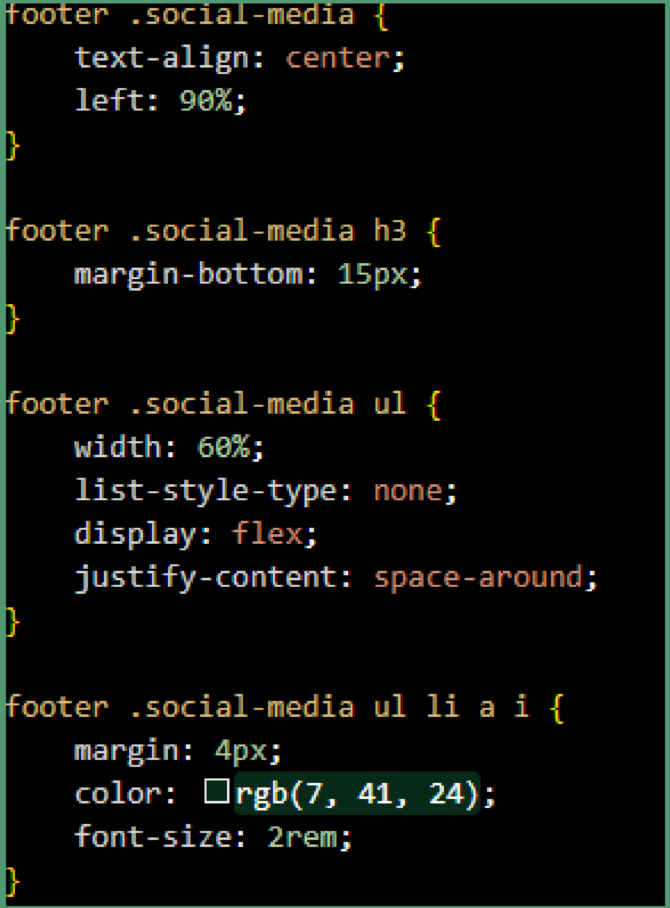
```html
<footer id="footer">
    <div class="background-img"><img src="zold.jpg" alt="zold"></div>
    <div class="tobbi">
    <div class="subscription">
        <h3>Don't miss out on our new games!</h3>
        <form action="/" method="post">
            <input type="text" id="name" name="user_name" value="Telefon">
            <input type="email" id="email" name="émail" value="Email">
            <button type="submit">Subscribe</button>
        </form>
    </div>
```



```css
footer {
    background: #F7FCF4;
    position: relative;
    overflow: hidden;
    height: 700px;
}
```

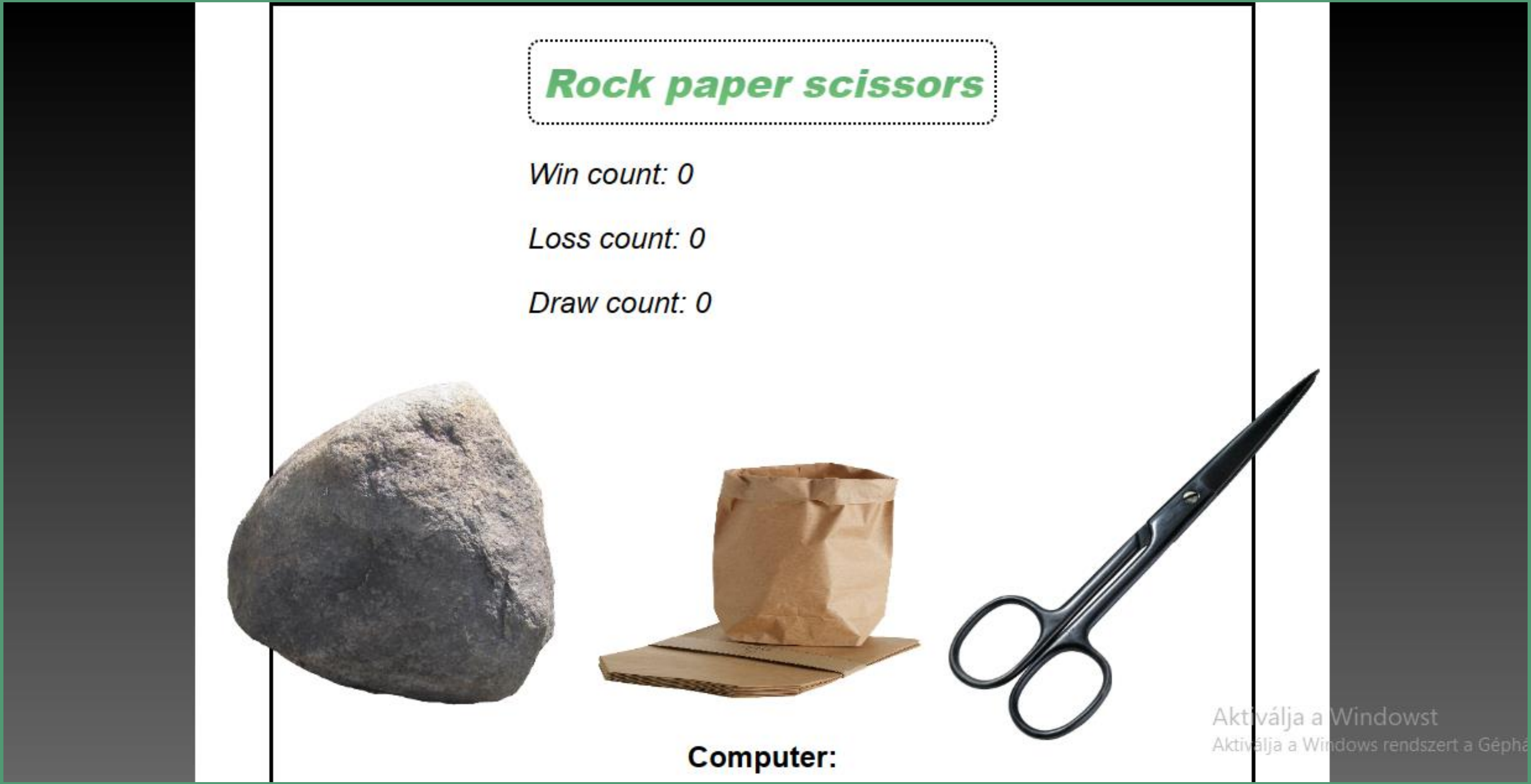


```css
.tobbi {
    position: absolute;
    top: 20%;
    /* left: 10%; */
    width: 100%;
    display: flex;
    justify-content: space-around;
    align-items: center;
}

footer .subscription {
    text-align: center;
    width: 40%;
    padding: 50px;
}
```



```css
footer .social-media {
    text-align: center;
    left: 90%;
}

footer .social-media h3 {
    margin-bottom: 15px;
}

footer .social-media ul {
    width: 60%;
    list-style-type: none;
    display: flex;
    justify-content: space-around;
}

footer .social-media ul li a i {
    margin: 4px;
    color: rgb(7, 41, 24);
    font-size: 2rem;
}
```

ROCK PAPER SCISSORS



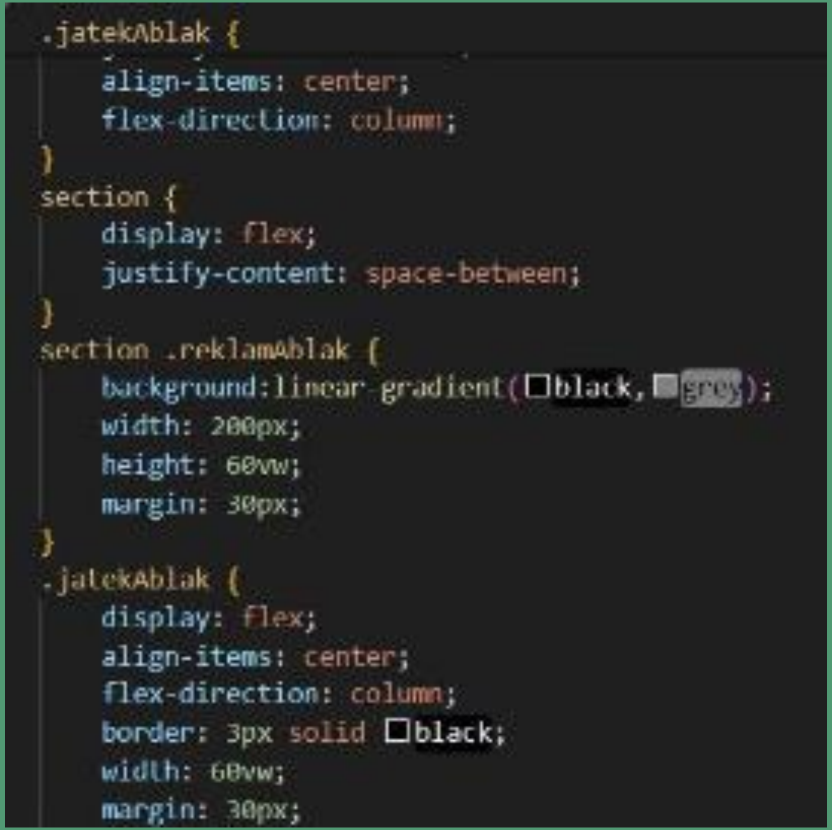
Rock paper scissors

Win count: 0

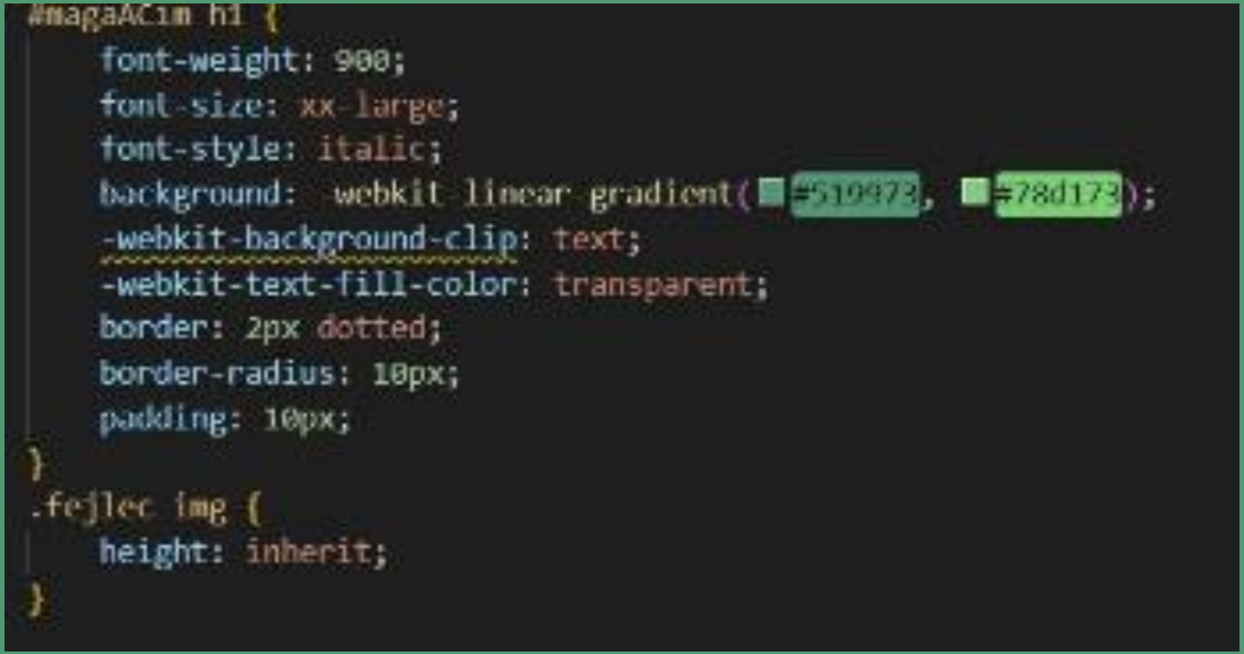
Loss count: 0

Draw count: 0

Computer:

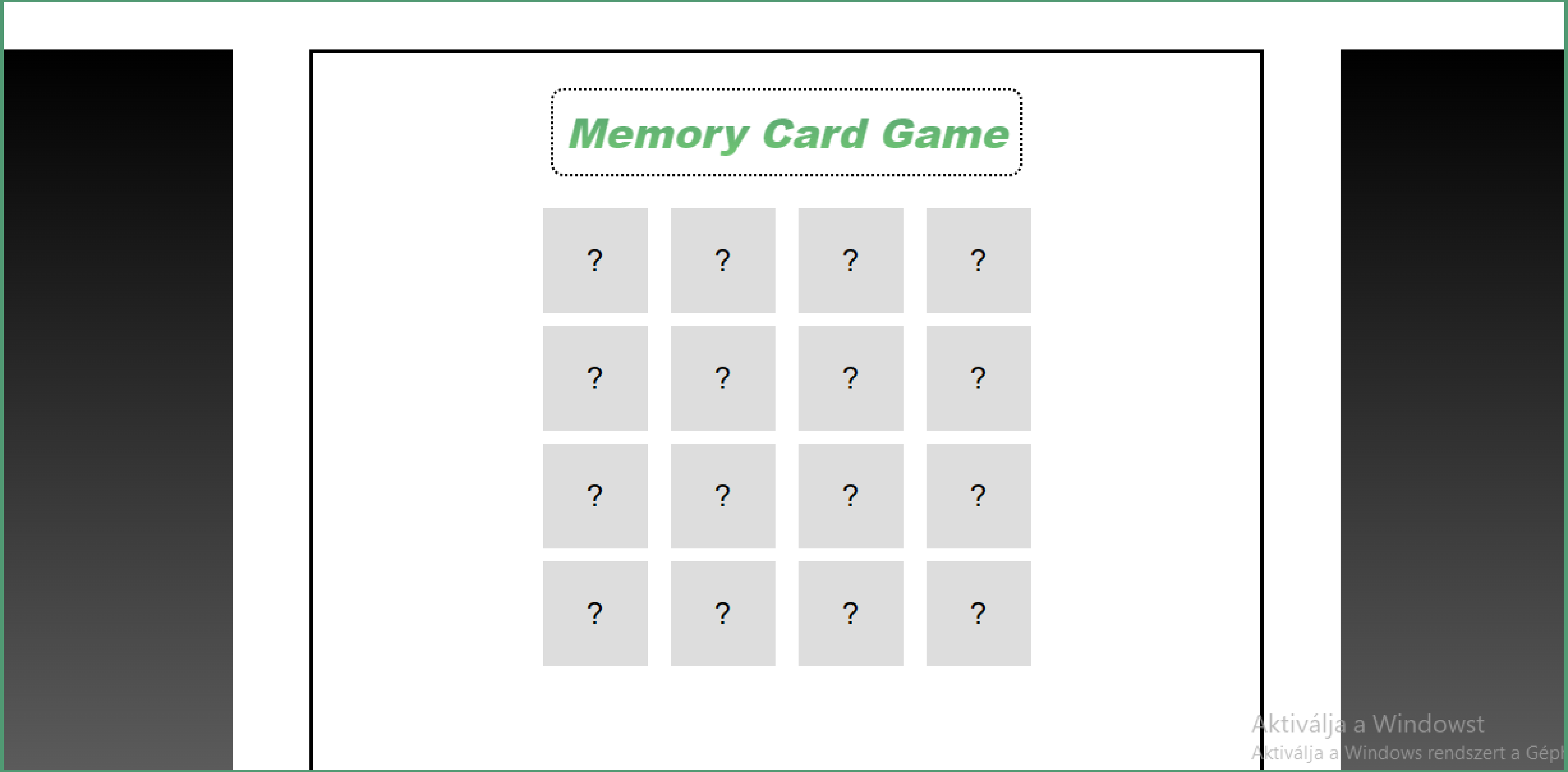


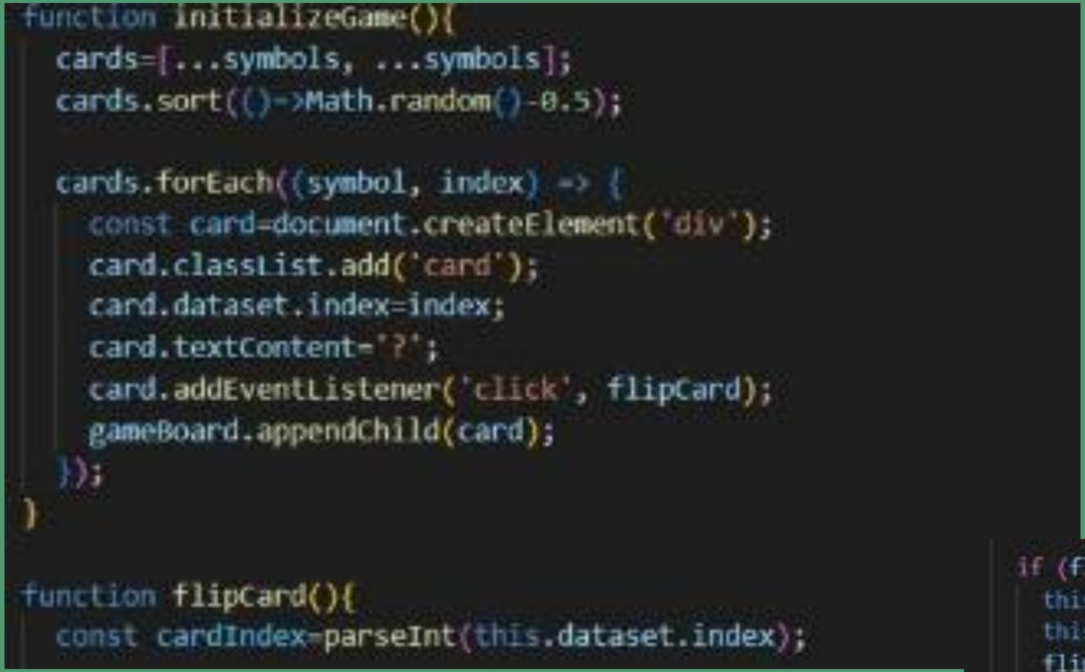
```css
.jatekAblak {
    align-items: center;
    flex-direction: column;
}
section {
    display: flex;
    justify-content: space-between;
}
section .reklamAblak {
    background:linear-gradient(black, grey);
    width: 200px;
    height: 60vw;
    margin: 30px;
}
.jatekAblak {
    display: flex;
    align-items: center;
    flex-direction: column;
    border: 3px solid black;
    width: 60vw;
    margin: 30px;
```



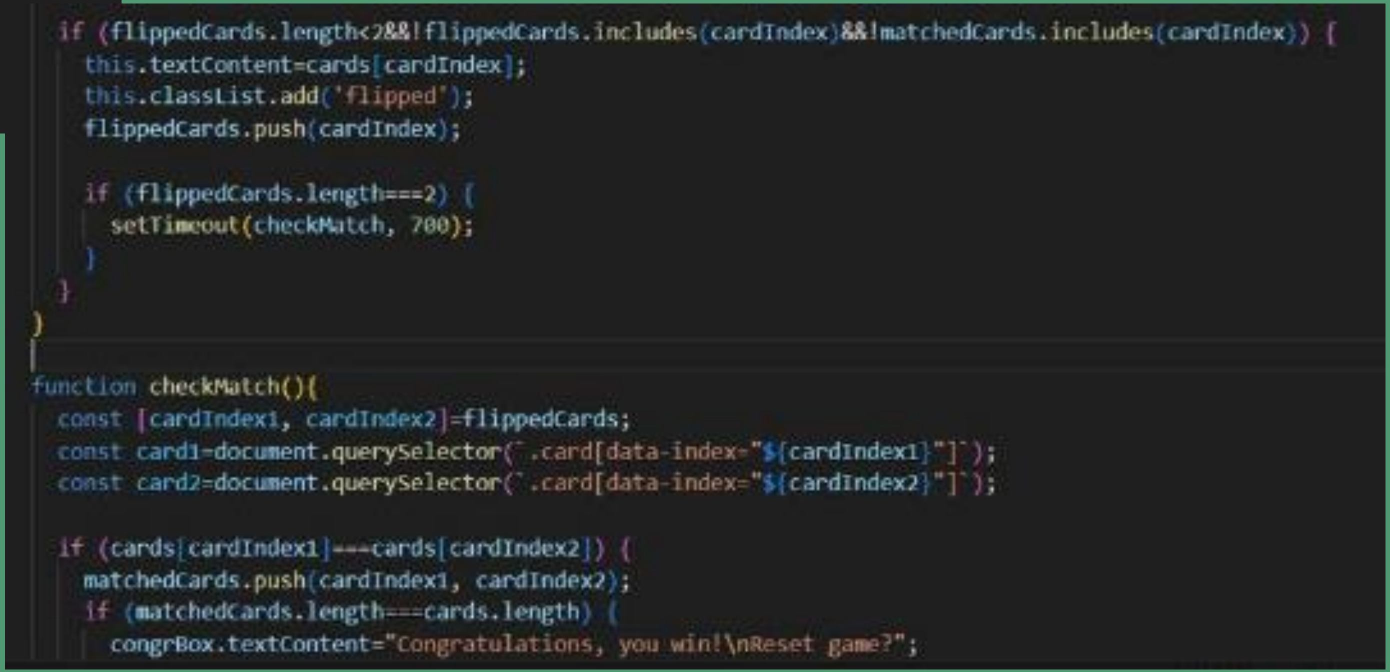
```css
#magaACim h1 {
    font-weight: 900;
    font-size: xx-large;
    font-style: italic;
    background: -webkit-linear-gradient(#519973, #78d173);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
    border: 2px dotted;
    border-radius: 10px;
    padding: 10px;
}
.fejlec img {
    height: inherit;
}
```

MEMORY CARD GAME



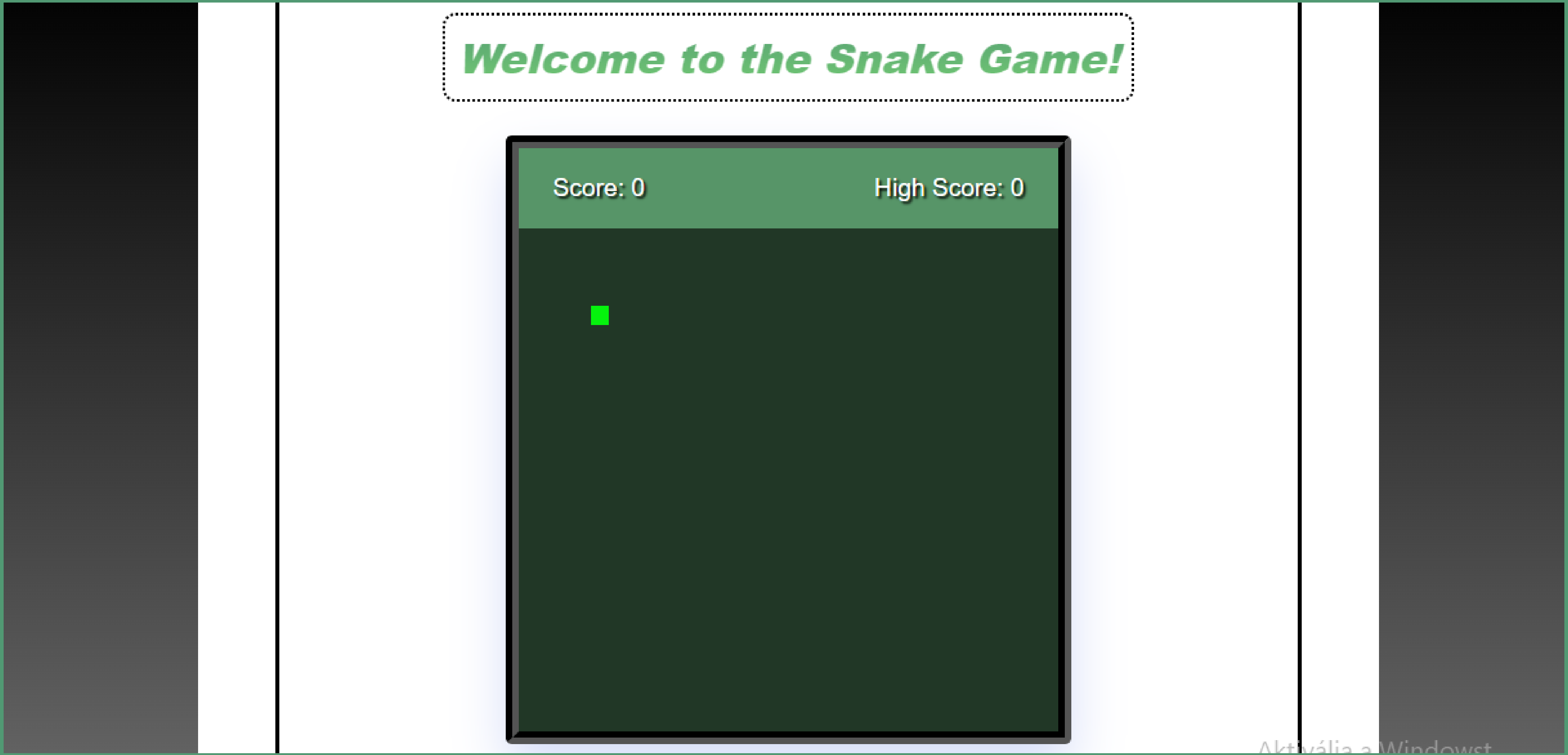


```
function initializeGame(){
  cards=[...symbols, ...symbols];
  cards.sort(()=>Math.random()-0.5);

  cards.forEach((symbol, index) => {
    const card=document.createElement('div');
    card.classList.add('card');
    card.dataset.index=index;
    card.textContent='?';
    card.addEventListener('click', flipCard);
    gameBoard.appendChild(card);
  });
}

function flipCard(){
  const cardIndex=parseInt(this.dataset.index);
```



```
  if (flippedCards.length<2&&!flippedCards.includes(cardIndex)&&!matchedCards.includes(cardIndex)) {
    this.textContent=cards[cardIndex];
    this.classList.add('flipped');
    flippedCards.push(cardIndex);

    if (flippedCards.length===2) {
      setTimeout(checkMatch, 700);
    }
  }
}

function checkMatch(){
  const [cardIndex1, cardIndex2]=flippedCards;
  const card1=document.querySelector(`.card[data-index="${cardIndex1}"]`);
  const card2=document.querySelector(`.card[data-index="${cardIndex2}"]`);

  if (cards[cardIndex1]===cards[cardIndex2]) {
    matchedCards.push(cardIndex1, cardIndex2);
    if (matchedCards.length===cards.length) {
      congrBox.textContent="Congratulations, you win!\nReset game?";
```

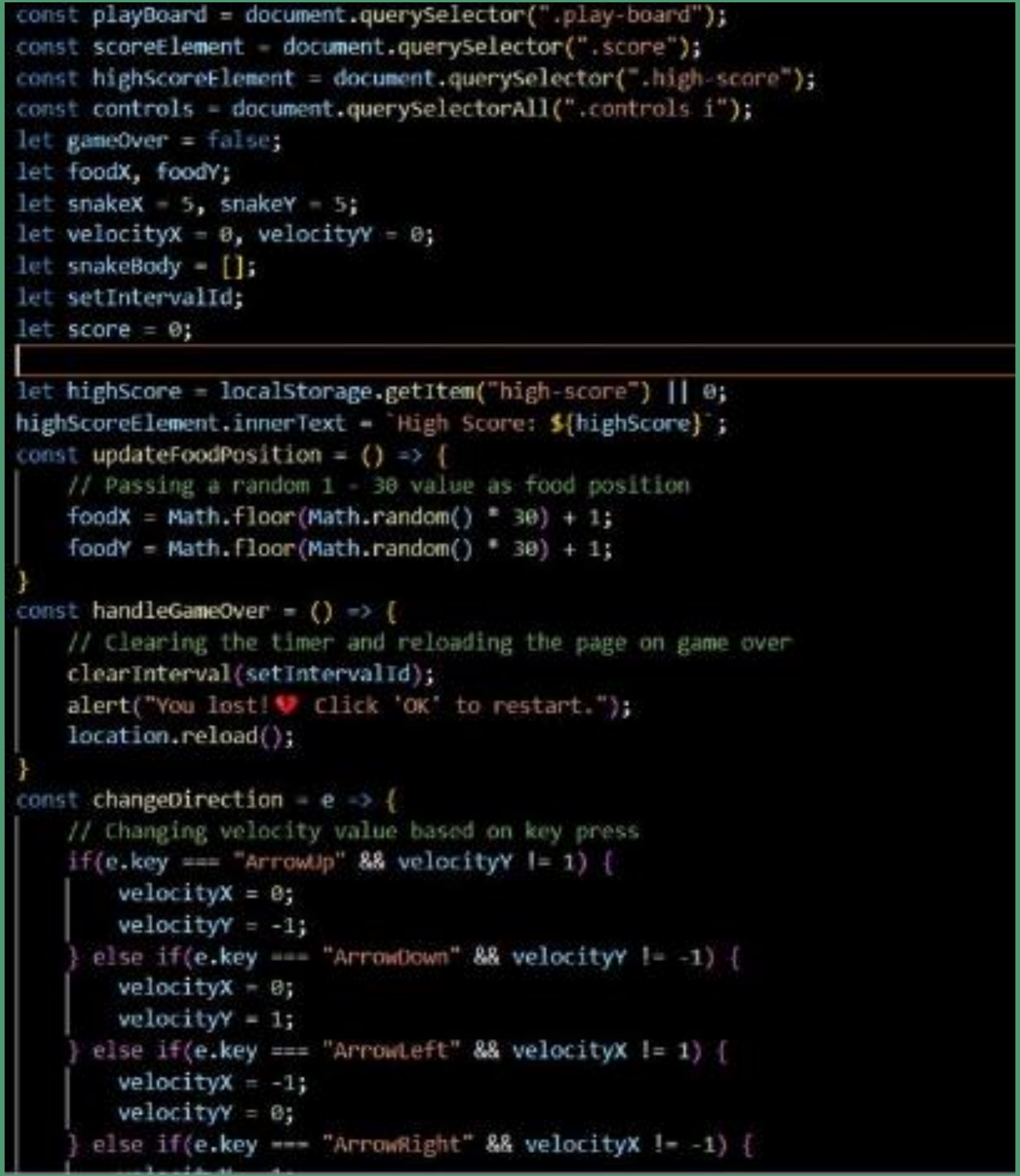
SNAKE GAME



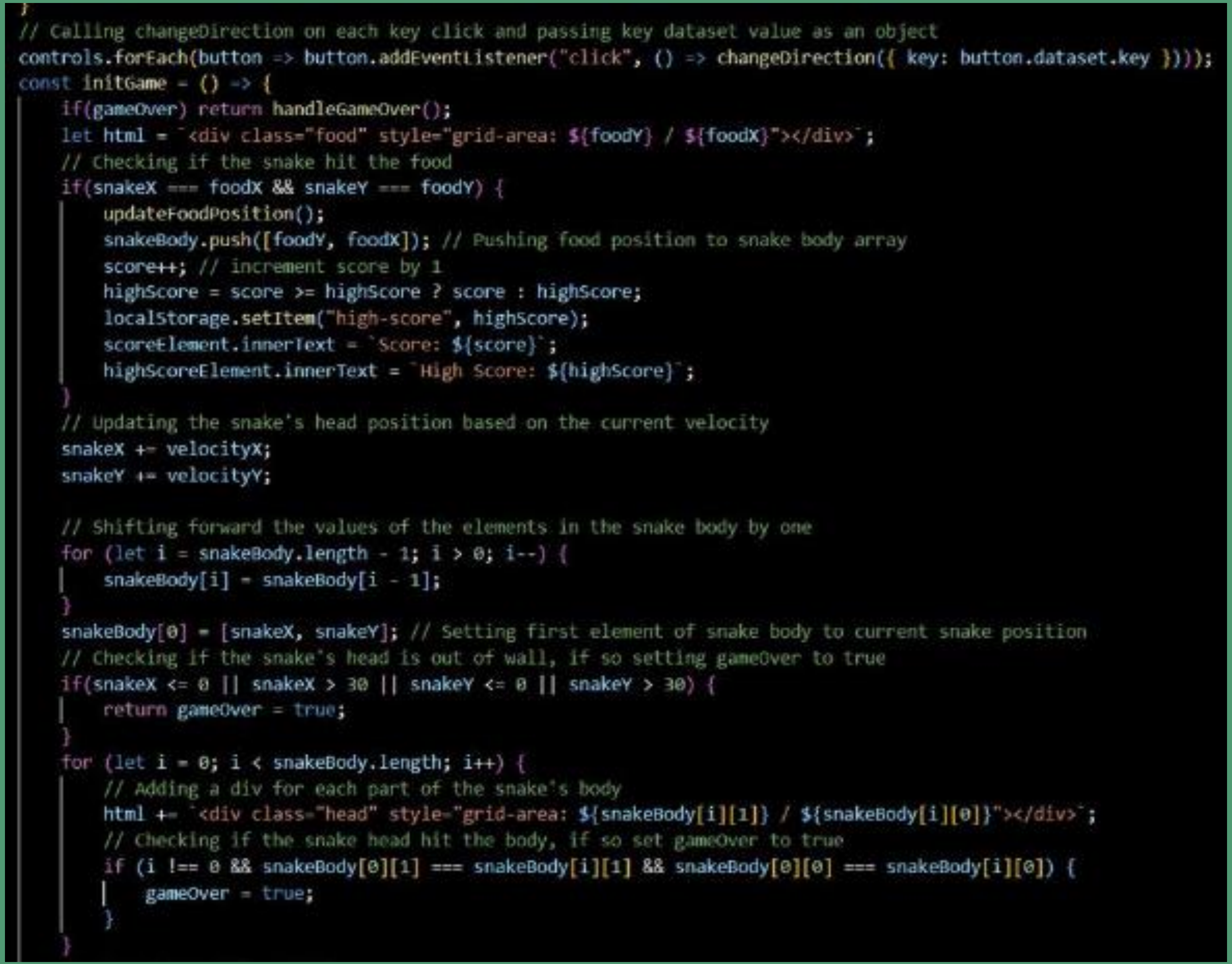
Welcome to the Snake Game!

Score: 0

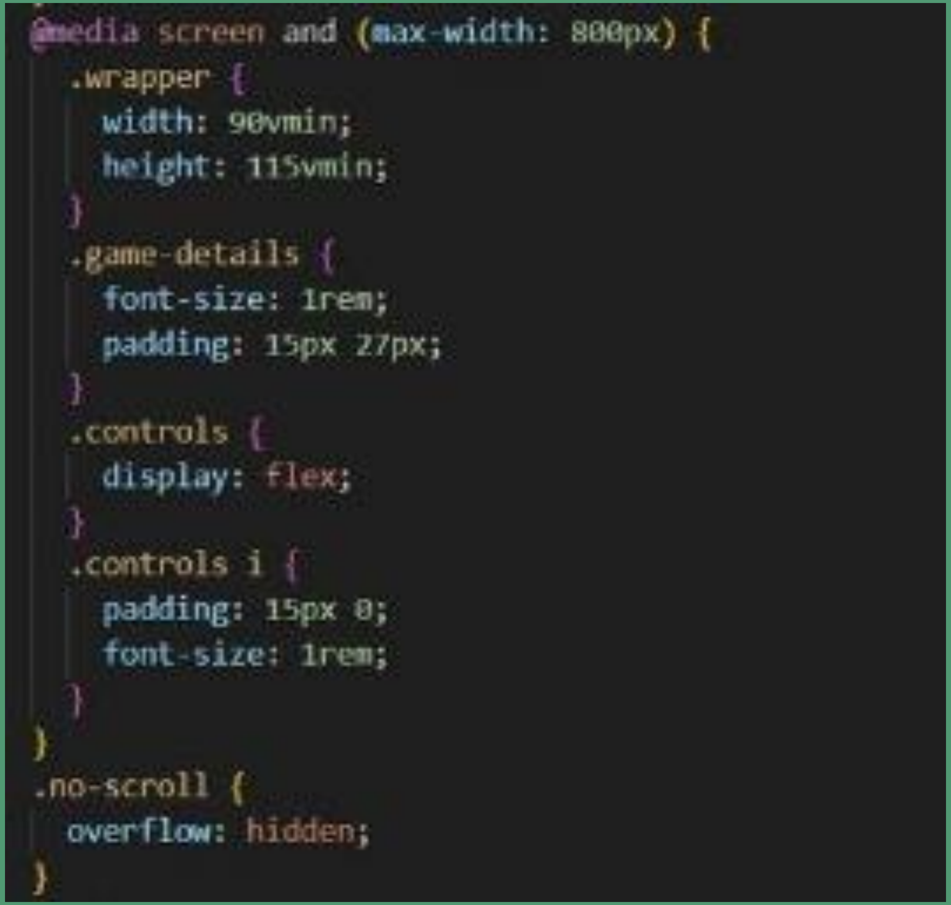
High Score: 0



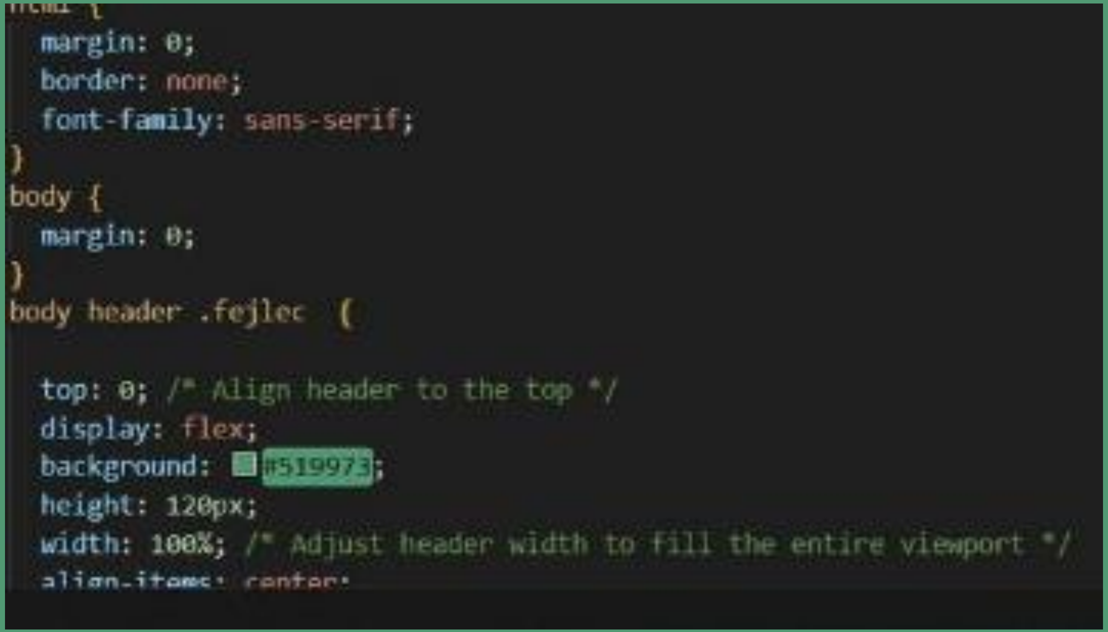
```
const playBoard = document.querySelector(".play-board");
const scoreElement = document.querySelector(".score");
const highScoreElement = document.querySelector(".high-score");
const controls = document.querySelectorAll(".controls i");
let gameOver = false;
let foodX, foodY;
let snakeX = 5, snakeY = 5;
let velocityX = 0, velocityY = 0;
let snakeBody = [];
let setIntervalId;
let score = 0;

let highScore = localStorage.getItem("high-score") || 0;
highScoreElement.innerText = `High Score: ${highScore}`;
const updateFoodPosition = () => {
    // Passing a random 1 - 30 value as food position
    foodX = Math.floor(Math.random() * 30) + 1;
    foodY = Math.floor(Math.random() * 30) + 1;
}
const handleGameOver = () => {
    // Clearing the timer and reloading the page on game over
    clearInterval(setIntervalId);
    alert("You lost!❤️ Click 'OK' to restart.");
    location.reload();
}
const changeDirection = e => {
    // Changing velocity value based on key press
    if(e.key === "ArrowUp" && velocityY != 1) {
        velocityX = 0;
        velocityY = -1;
    } else if(e.key === "ArrowDown" && velocityY != -1) {
        velocityX = 0;
        velocityY = 1;
    } else if(e.key === "ArrowLeft" && velocityX != 1) {
        velocityX = -1;
        velocityY = 0;
    } else if(e.key === "ArrowRight" && velocityX != -1) {
```



```
}
// Calling changeDirection on each key click and passing key dataset value as an object
controls.forEach(button => button.addEventListener("click", () => changeDirection({ key: button.dataset.key })));
const initGame = () => {
    if(gameOver) return handleGameOver();
    let html = `<div class="food" style="grid-area: ${foodY} / ${foodX}"></div>`;
    // Checking if the snake hit the food
    if(snakeX === foodX && snakeY === foodY) {
        updateFoodPosition();
        snakeBody.push([foodY, foodX]); // Pushing food position to snake body array
        score++; // increment score by 1
        highScore = score >= highScore ? score : highScore;
        localStorage.setItem("high-score", highScore);
        scoreElement.innerText = `Score: ${score}`;
        highScoreElement.innerText = `High Score: ${highScore}`;
    }
    // Updating the snake's head position based on the current velocity
    snakeX += velocityX;
    snakeY += velocityY;

    // Shifting forward the values of the elements in the snake body by one
    for (let i = snakeBody.length - 1; i > 0; i--) {
        snakeBody[i] = snakeBody[i - 1];
    }
    snakeBody[0] = [snakeX, snakeY]; // Setting first element of snake body to current snake position
    // Checking if the snake's head is out of wall, if so setting gameOver to true
    if(snakeX <= 0 || snakeX > 30 || snakeY <= 0 || snakeY > 30) {
        return gameOver = true;
    }
    for (let i = 0; i < snakeBody.length; i++) {
        // Adding a div for each part of the snake's body
        html += `<div class="head" style="grid-area: ${snakeBody[i][1]} / ${snakeBody[i][0]}"></div>`;
        // Checking if the snake head hit the body, if so set gameOver to true
        if (i !== 0 && snakeBody[0][1] === snakeBody[i][1] && snakeBody[0][0] === snakeBody[i][0]) {
            gameOver = true;
        }
    }
```

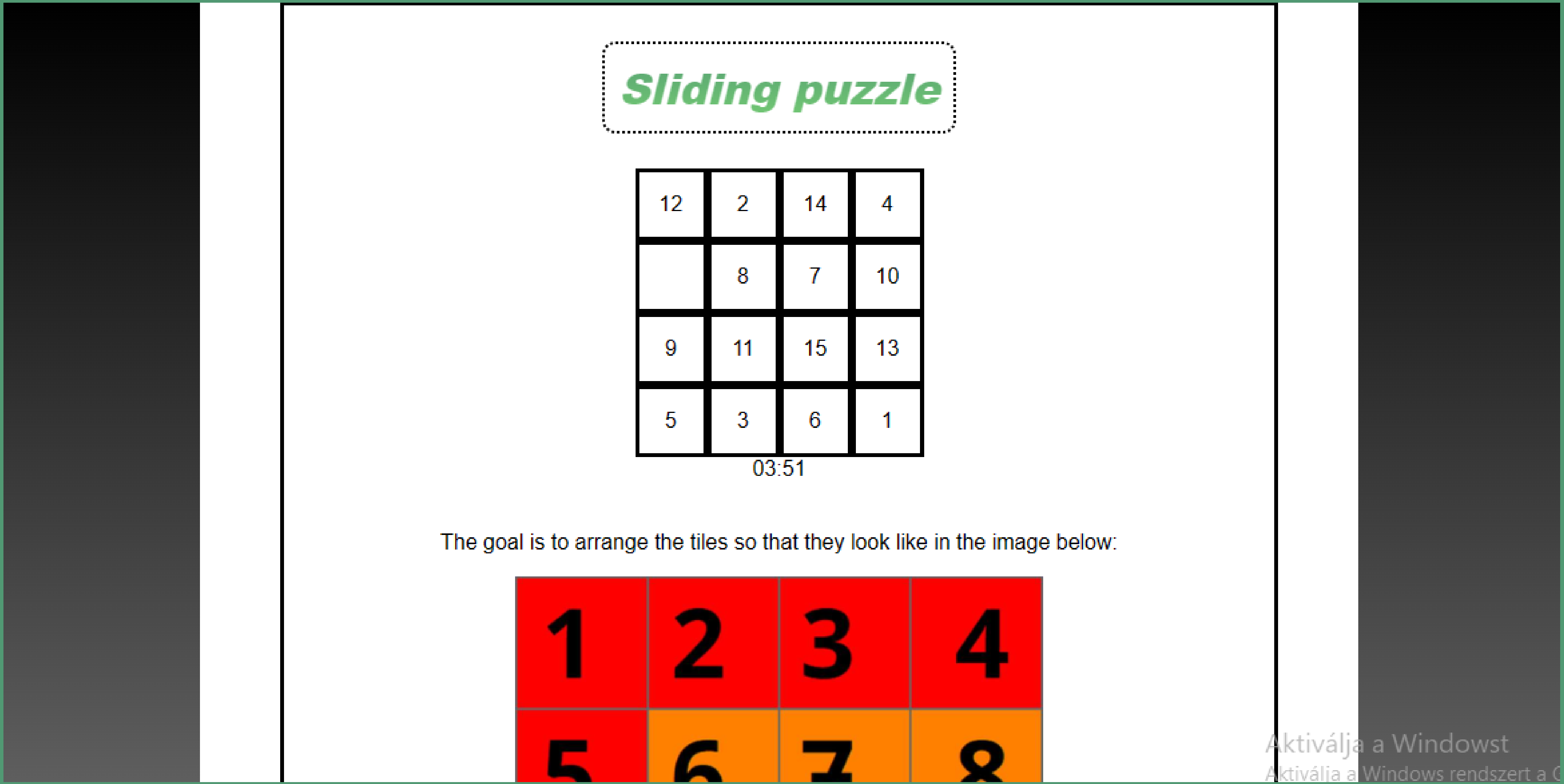


```css
@media screen and (max-width: 800px) {
  .wrapper {
    width: 90vmin;
    height: 115vmin;
  }
  .game-details {
    font-size: 1rem;
    padding: 15px 27px;
  }
  .controls {
    display: flex;
  }
  .controls i {
    padding: 15px 0;
    font-size: 1rem;
  }
}
.no-scroll {
  overflow: hidden;
}
```



```css
html {
  margin: 0;
  border: none;
  font-family: sans-serif;
}
body {
  margin: 0;
}
body header .fejlec  {

  top: 0; /* Align header to the top */
  display: flex;
  background: #519973;
  height: 120px;
  width: 100%; /* Adjust header width to fill the entire viewport */
  align-items: center;
```

SLIDING PUZZLE



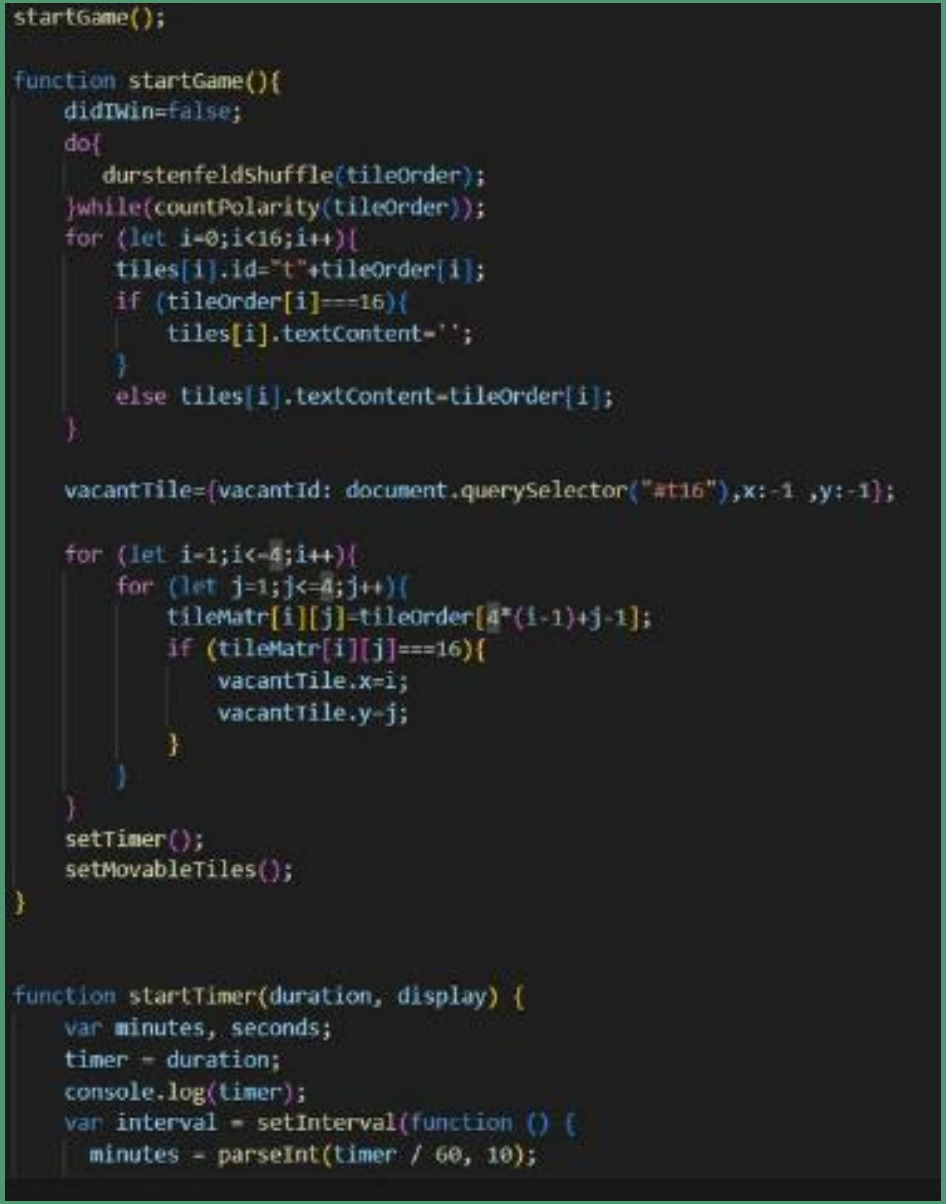
Sliding puzzle

12	2	14	4
	8	7	10
9	11	15	13
5	3	6	1

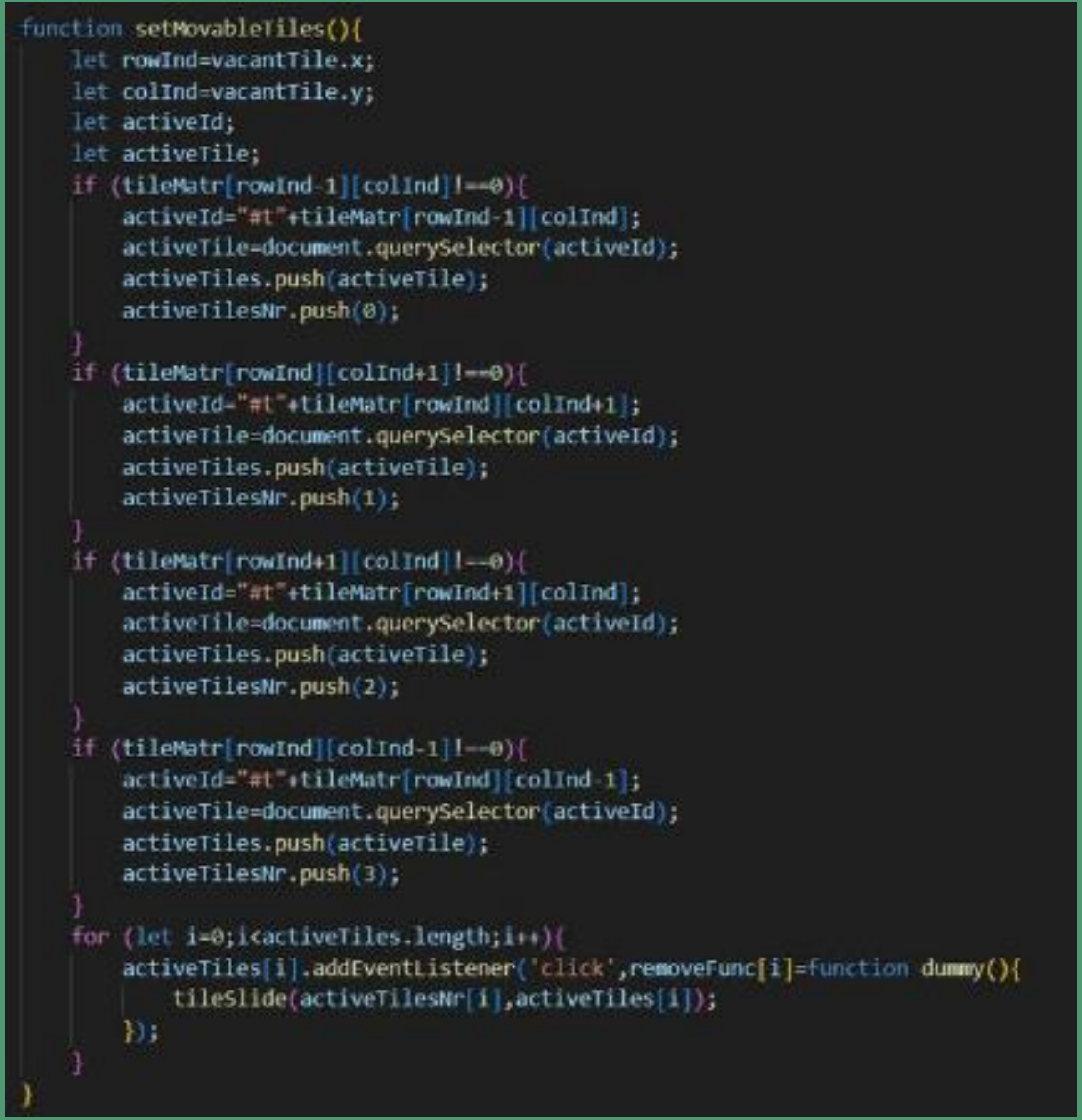
03:51

The goal is to arrange the tiles so that they look like in the image below:

1	2	3	4
5	6	7	8

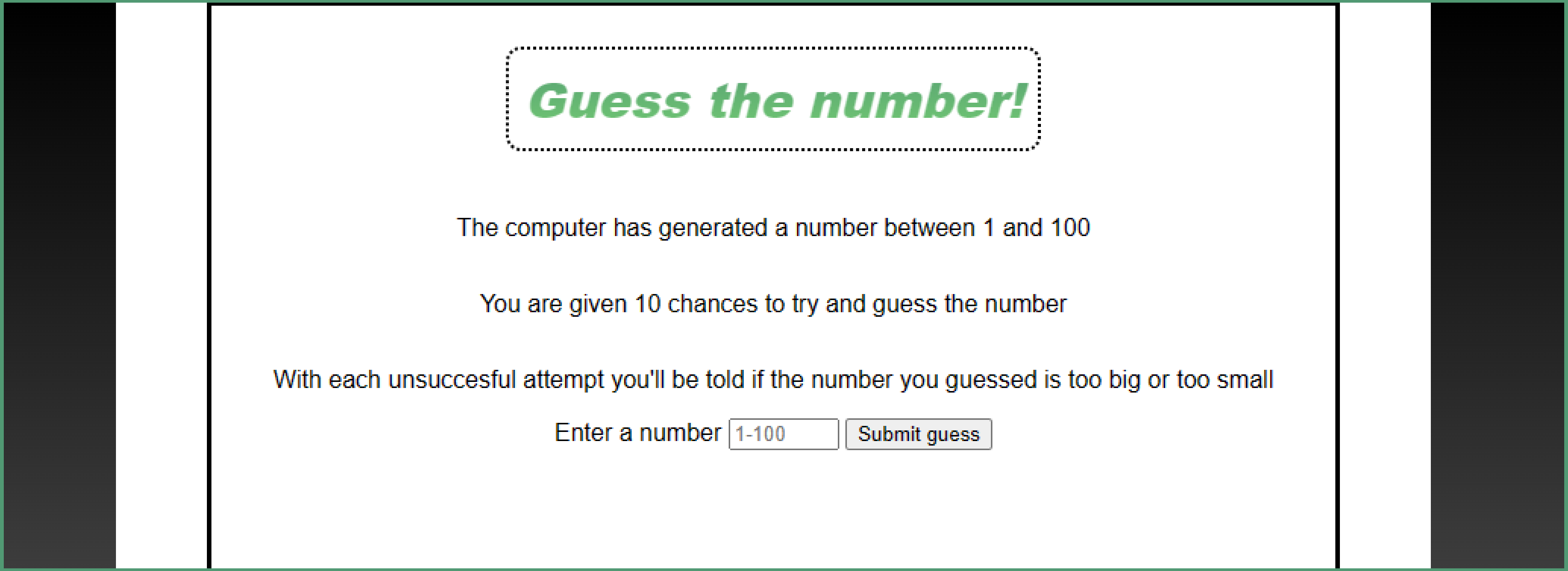


```
startGame();

function startGame(){
    didIWin=false;
    do{
        durstenfeldShuffle(tileOrder);
    }while(countPolarity(tileOrder));
    for (let i=0;i<16;i++){
        tiles[i].id="t"+tileOrder[i];
        if (tileOrder[i]===16){
            tiles[i].textContent='';
        }
        else tiles[i].textContent=tileOrder[i];
    }

    vacantTile={vacantId: document.querySelector("#t16"),x:-1 ,y:-1};

    for (let i=1;i<=4;i++){
        for (let j=1;j<=4;j++){
            tileMatr[i][j]=tileOrder[4*(i-1)+j-1];
            if (tileMatr[i][j]===16){
                vacantTile.x=i;
                vacantTile.y=j;
            }
        }
    }
    setTimer();
    setMovableTiles();
}


function startTimer(duration, display) {
    var minutes, seconds;
    timer = duration;
    console.log(timer);
    var interval = setInterval(function () {
      minutes = parseInt(timer / 60, 10);
```



```
function setMovableTiles(){
    let rowInd=vacantTile.x;
    let colInd=vacantTile.y;
    let activeId;
    let activeTile;
    if (tileMatr[rowInd-1][colInd]!==0){
        activeId="#t"+tileMatr[rowInd-1][colInd];
        activeTile=document.querySelector(activeId);
        activeTiles.push(activeTile);
        activeTilesNr.push(0);
    }
    if (tileMatr[rowInd][colInd+1]!==0){
        activeId="#t"+tileMatr[rowInd][colInd+1];
        activeTile=document.querySelector(activeId);
        activeTiles.push(activeTile);
        activeTilesNr.push(1);
    }
    if (tileMatr[rowInd+1][colInd]!==0){
        activeId="#t"+tileMatr[rowInd+1][colInd];
        activeTile=document.querySelector(activeId);
        activeTiles.push(activeTile);
        activeTilesNr.push(2);
    }
    if (tileMatr[rowInd][colInd-1]!==0){
        activeId="#t"+tileMatr[rowInd][colInd-1];
        activeTile=document.querySelector(activeId);
        activeTiles.push(activeTile);
        activeTilesNr.push(3);
    }
    for (let i=0;i<activeTiles.length;i++){
        activeTiles[i].addEventListener('click',removeFunc[i]=function dummy(){
            tileSlide(activeTilesNr[i],activeTiles[i]);
        });
    }
}
```

GUESS THE NUMBER

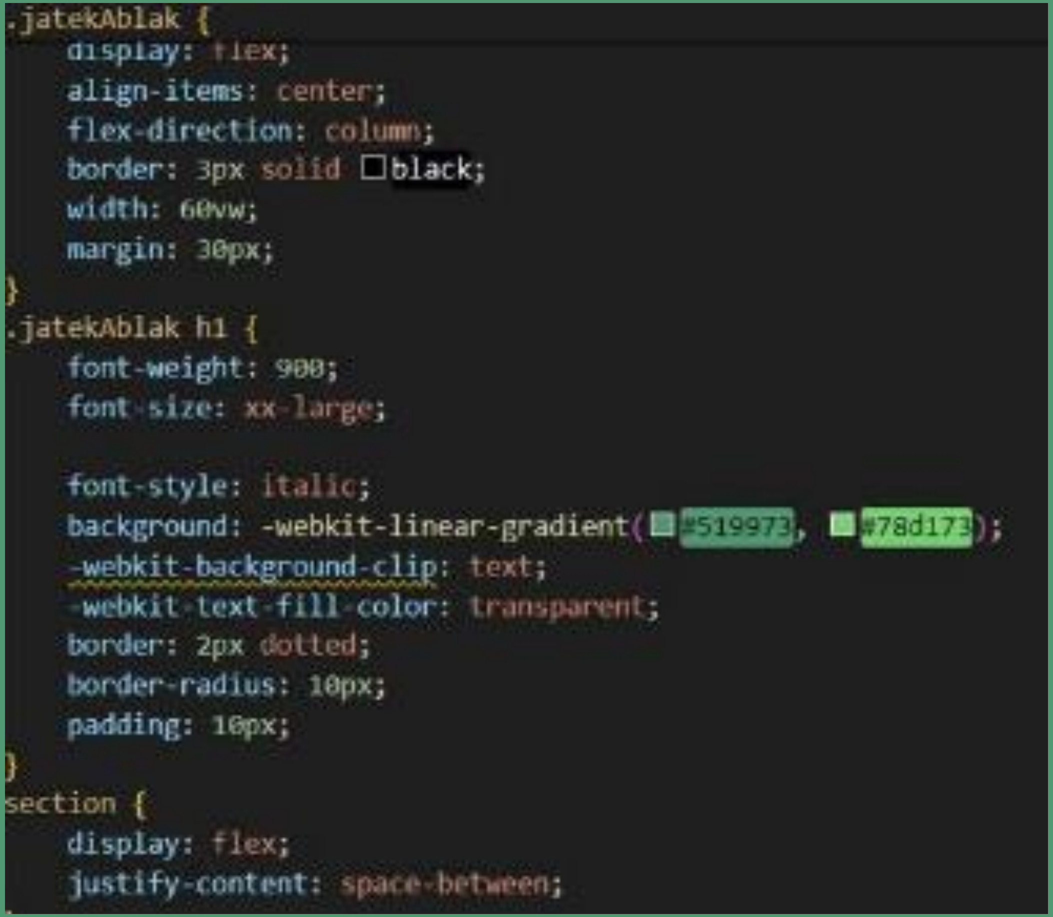


Guess the number!

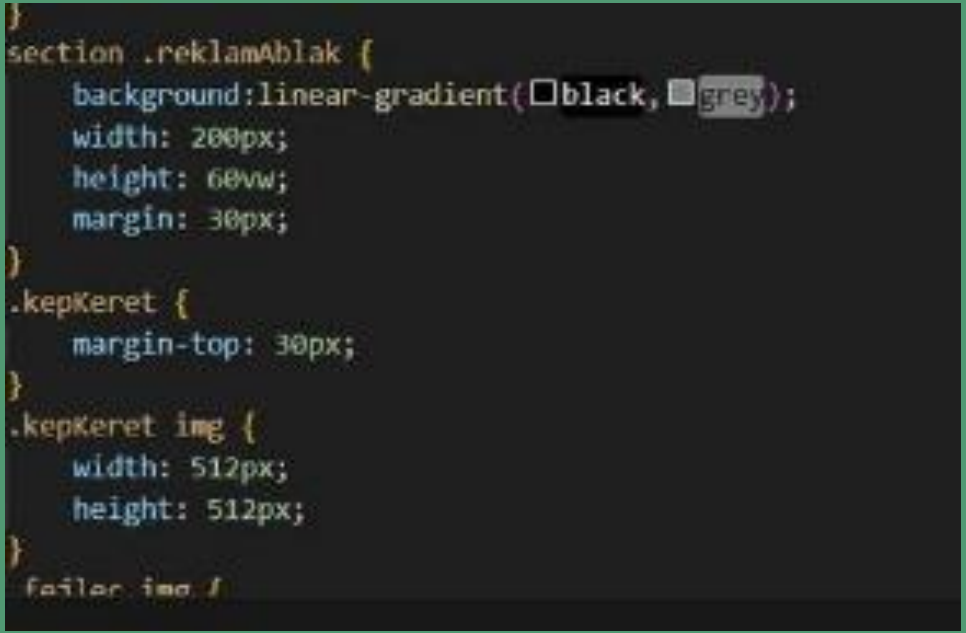
The computer has generated a number between 1 and 100

You are given 10 chances to try and guess the number

With each unsuccesful attempt you'll be told if the number you guessed is too big or too small



```css
.jatekAblak {
    display: flex;
    align-items: center;
    flex-direction: column;
    border: 3px solid black;
    width: 60vw;
    margin: 30px;
}
.jatekAblak h1 {
    font-weight: 900;
    font-size: xx-large;

    font-style: italic;
    background: -webkit-linear-gradient(#519973, #78d173);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
    border: 2px dotted;
    border-radius: 10px;
    padding: 10px;
}
section {
    display: flex;
    justify-content: space-between;
```



```css
}
section .reklamAblak {
    background:linear-gradient(black, grey);
    width: 200px;
    height: 60vw;
    margin: 30px;
}
.kepKeret {
    margin-top: 30px;
}
.kepKeret img {
    width: 512px;
    height: 512px;
}
.failer img {
```

THANK YOU FOR YOUR ATTENCION!