3 Game html, css, dan Javascript

1. Word Scramble Game

Index.html

```
<!DOCTYPE html>
<!-- Coding By CodingNepal - youtube.com/codingnepal -->
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Word Scramble Game | CodingNepal</title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <script src="js/words.js" defer></script>
    <script src="js/script.js" defer></script>
  </head>
  <body>
    <div class="container">
        <h2>Word Scramble</h2>
        <div class="content">
            <p class="word"></p>
            <div class="details">
                <p class="hint">Hint: <span></span></p>
                <p class="time">Time Left:
<span><b>30</b>s</span></p>
            </div>
            <input type="text" spellcheck="false" placeholder="Enter
a valid word">
            <div class="buttons">
                <button class="refresh-word">Refresh Word</button>
                <button class="check-word">Check Word</button>
            </div>
```

```
        </div>
    </div>
  </body>
</html>
```

Style.css

```
/* Import Google font - Poppins */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;6
00&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  display: flex;
  padding: 0 10px;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  background: #5372F0;
}
.container{
  width: 440px;
  border-radius: 7px;
  background: #fff;
  box-shadow: 0 10px 20px rgba(0,0,0,0.08);
}
.container h2{
  font-size: 25px;
  font-weight: 500;
  padding: 16px 25px;
  border-bottom: 1px solid #ccc;
}
.container .content{
  margin: 25px 20px 35px;
}
.content .word{
  user-select: none;
  font-size: 33px;
  font-weight: 500;
  text-align: center;
  letter-spacing: 24px;
  margin-right: -24px;
  word-break: break-all;
  text-transform: uppercase;
```

```
}
.content .details{
  margin: 25px 0 20px;
}
.details p{
  font-size: 18px;
  margin-bottom: 10px;
}
.details p b{
  font-weight: 500;
}
.content input{
  width: 100%;
  height: 60px;
  outline: none;
  padding: 0 16px;
  font-size: 18px;
  border-radius: 5px;
  border: 1px solid #bfbfbf;
}
.content input:focus{
  box-shadow: 0px 2px 4px rgba(0,0,0,0.08);
}
.content input::placeholder{
  color: #aaa;
}
.content input:focus::placeholder{
  color: #bfbfbf;
}
.content .buttons{
  display: flex;
  margin-top: 20px;
  justify-content: space-between;
}
.buttons button{
  border: none;
  outline: none;
  color: #fff;
  cursor: pointer;
  padding: 15px 0;
  font-size: 17px;
  border-radius: 5px;
  width: calc(100% / 2 - 8px);
  transition: all 0.3s ease;
}
.buttons button:active{
  transform: scale(0.97);
}
```

```css
.buttons .refresh-word{
  background: #6C757D;
}
.buttons .refresh-word:hover{
  background: #5f666d;
}
.buttons .check-word{
  background: #5372F0;
}
.buttons .check-word:hover{
  background: #2c52ed;
}

@media screen and (max-width: 470px) {
  .container h2{
    font-size: 22px;
    padding: 13px 20px;
  }
  .content .word{
    font-size: 30px;
    letter-spacing: 20px;
    margin-right: -20px;
  }
  .container .content{
    margin: 20px 20px 30px;
  }
  .details p{
    font-size: 16px;
    margin-bottom: 8px;
  }
  .content input{
    height: 55px;
    font-size: 17px;
  }
  .buttons button{
    padding: 14px 0;
    font-size: 16px;
    width: calc(100% / 2 - 7px);
  }
}
```

Javascript

- Script.js

```javascript
const wordText = document.querySelector(".word"),
hintText = document.querySelector(".hint span"),
timeText = document.querySelector(".time b"),
inputField = document.querySelector("input"),
refreshBtn = document.querySelector(".refresh-word"),
```

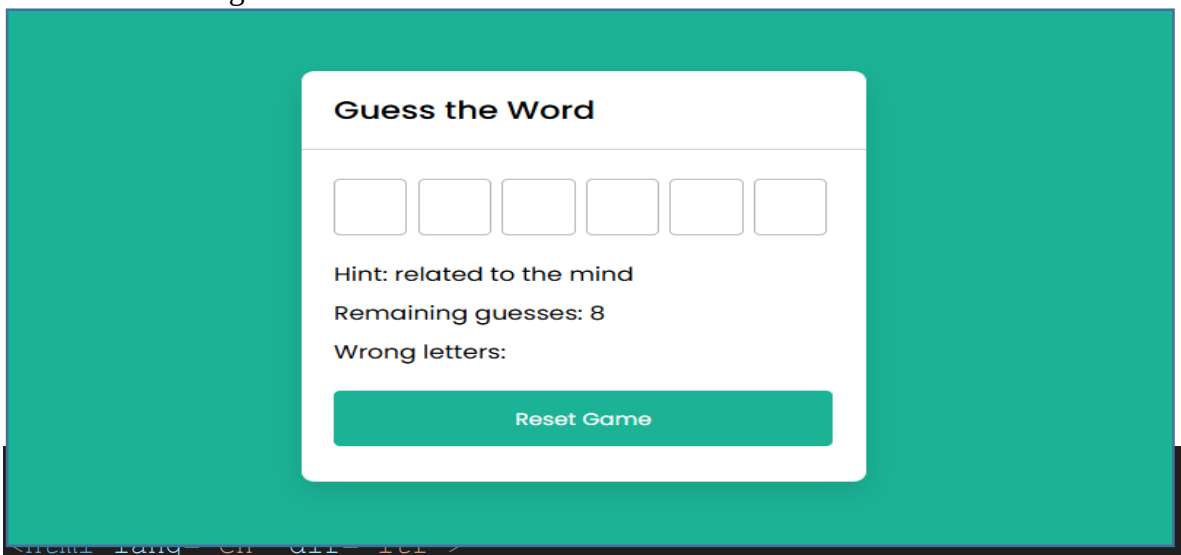
```
checkBtn = document.querySelector(".check-word");

let correctWord, timer;

const initTimer = maxTime => {
    clearInterval(timer);
    timer = setInterval(() => {
        if(maxTime > 0) {
            maxTime--;
            return timeText.innerText = maxTime;
        }
        alert(`Time off! ${correctWord.toUpperCase()} was the
correct word`);
        initGame();
    }, 1000);
}

const initGame = () => {
    initTimer(30);
    let randomObj = words[Math.floor(Math.random() * words.length)];
    let wordArray = randomObj.word.split("");
    for (let i = wordArray.length - 1; i > 0; i--) {
        let j = Math.floor(Math.random() * (i + 1));
        [wordArray[i], wordArray[j]] = [wordArray[j], wordArray[i]];
    }
    wordText.innerText = wordArray.join("");
    hintText.innerText = randomObj.hint;
    correctWord = randomObj.word.toLowerCase();;
    inputField.value = "";
    inputField.setAttribute("maxlength", correctWord.length);
}
initGame();

const checkWord = () => {
    let userWord = inputField.value.toLowerCase();
    if(!userWord) return alert("Please enter the word to check!");
    if(userWord !== correctWord) return alert(`Oops! ${userWord} is
not a correct word`);
    alert(`Congrats! ${correctWord.toUpperCase()} is the correct
word`);
    initGame();
}

refreshBtn.addEventListener("click", initGame);
checkBtn.addEventListener("click", checkWord);
```

words.js

```
let words = [
    {
        word: "addition",
        hint: "The process of adding numbers"
    },
    {
        word: "meeting",
        hint: "Event in which people come together"
    },
    {
        word: "number",
        hint: "Math symbol used for counting"
    },
    {
        word: "exchange",
        hint: "The act of trading"
    },
    {
        word: "canvas",
        hint: "Piece of fabric for oil painting"
    },
    {
        word: "garden",
        hint: "Space for planting flower and plant"
    },
    {
        word: "position",
        hint: "Location of someone or something"
    },
    {
        word: "feather",
        hint: "Hair like outer covering of bird"
    },
    {
        word: "comfort",
        hint: "A pleasant feeling of relaxation"
    },
    {
        word: "tongue",
        hint: "The muscular organ of mouth"
    },
    {
        word: "expansion",
        hint: "The process of increase or grow"
    },
    {
        word: "country",
        hint: "A politically identified region"
```

```
    },
    {
        word: "group",
        hint: "A number of objects or persons"
    },
    {
        word: "taste",
        hint: "Ability of tongue to detect flavour"
    },
    {
        word: "store",
        hint: "Large shop where goods are traded"
    },
    {
        word: "field",
        hint: "Area of land for farming activities"
    },
    {
        word: "friend",
        hint: "Person other than a family member"
    },
    {
        word: "pocket",
        hint: "A bag for carrying small items"
    },
    {
        word: "needle",
        hint: "A thin and sharp metal pin"
    },
    {
        word: "expert",
        hint: "Person with extensive knowledge"
    },
    {
        word: "statement",
        hint: "A declaration of something"
    },
    {
        word: "second",
        hint: "One-sixtieth of a minute"
    },
    {
        word: "library",
        hint: "Place containing collection of books"
    },
]
```

2. Word Guessing Game




```
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Guess a Word Game JavaScript | CodingNepal</title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  </head>
  <body>
    <div class="wrapper">
      <h1>Guess the Word</h1>
      <div class="content">
        <input type="text" class="typing-input" maxlength="1">
        <div class="inputs"></div>
        <div class="details">
          <p class="hint">Hint: <span></span></p>
          <p class="guess-left">Remaining guesses: <span></span></p>
          <p class="wrong-letter">Wrong letters: <span></span></p>
        </div>
        <button class="reset-btn">Reset Game</button>
      </div>
    </div>

    <script src="js/words.js"></script>
    <script src="js/script.js"></script>

  </body>
</html>
```

Style.css

```
/* Import Google font - Poppins */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;6
00&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  display: flex;
  padding: 0 10px;
  min-height: 100vh;
  align-items: center;
  justify-content: center;
  background: #1BB295;
}
.wrapper{
  width: 430px;
  background: #fff;
  border-radius: 10px;
  box-shadow: 0 10px 25px rgba(0,0,0,0.1);
}
.wrapper h1{
  font-size: 25px;
  font-weight: 500;
  padding: 20px 25px;
  border-bottom: 1px solid #ccc;
}
.wrapper .content{
  margin: 25px 25px 35px;
}
.content .inputs{
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
}
.inputs input{
  height: 57px;
  width: 56px;
  margin: 4px;
  font-size: 24px;
  font-weight: 500;
  color: #1ba98c;
  text-align: center;
  border-radius: 5px;
```

```
  background: none;
  pointer-events: none;
  text-transform: uppercase;
  border: 1px solid #B5B5B5;
}
.typing-input {
  opacity: 0;
  z-index: -999;
  position: absolute;
  pointer-events: none;
}
.inputs input:first-child{
  margin-left: 0px;
}
.content .details{
  margin: 20px 0 25px;
}
.details p{
  font-size: 19px;
  margin-bottom: 10px;
}
.content .reset-btn{
  width: 100%;
  border: none;
  cursor: pointer;
  color: #fff;
  outline: none;
  padding: 15px 0;
  font-size: 17px;
  border-radius: 5px;
  background: #1BB295;
  transition: all 0.3s ease;
}
.content .reset-btn:hover{
  background: #18a589;
}

@media screen and (max-width: 460px) {
  .wrapper {
    width: 100%;
  }
  .wrapper h1{
    font-size: 22px;
    padding: 16px 20px;
  }
  .wrapper .content{
    margin: 25px 20px 35px;
  }
```

```
  .inputs input{
    height: 51px;
    width: 50px;
    margin: 3px;
    font-size: 22px;
  }
  .details p{
    font-size: 17px;
  }
  .content .reset-btn{
    padding: 14px 0;
    font-size: 16px;
  }
}
```

Javascript
Script.js

```
const inputs = document.querySelector(".inputs"),
hintTag = document.querySelector(".hint span"),
guessLeft = document.querySelector(".guess-left span"),
wrongLetter = document.querySelector(".wrong-letter span"),
resetBtn = document.querySelector(".reset-btn"),
typingInput = document.querySelector(".typing-input");

let word, maxGuesses, incorrectLetters = [], correctLetters = [];

function randomWord() {
    let ranItem = wordList[Math.floor(Math.random() *
wordList.length)];
    word = ranItem.word;
    maxGuesses = word.length >= 5 ? 8 : 6;
    correctLetters = []; incorrectLetters = [];
    hintTag.innerText = ranItem.hint;
    guessLeft.innerText = maxGuesses;
    wrongLetter.innerText = incorrectLetters;

    let html = "";
    for (let i = 0; i < word.length; i++) {
        html += `<input type="text" disabled>`;
        inputs.innerHTML = html;
    }
}
randomWord();

function initGame(e) {
    let key = e.target.value.toLowerCase();
    if(key.match(/^[A-Za-z]+$/) && !incorrectLetters.includes(`
${key}`) && !correctLetters.includes(key)) {
```

```
        if(word.includes(key)) {
            for (let i = 0; i < word.length; i++) {
                if(word[i] == key) {
                    correctLetters += key;
                    inputs.querySelectorAll("input")[i].value = key;
                }
            }
        } else {
            maxGuesses--;
            incorrectLetters.push(` ${key}`);
        }
        guessLeft.innerText = maxGuesses;
        wrongLetter.innerText = incorrectLetters;
    }
    typingInput.value = "";

    setTimeout(() => {
        if(correctLetters.length === word.length) {
            alert(`Congrats! You found the word
${word.toUpperCase()}`);
            return randomWord();
        } else if(maxGuesses < 1) {
            alert("Game over! You don't have remaining guesses");
            for(let i = 0; i < word.length; i++) {
                inputs.querySelectorAll("input")[i].value = word[i];
            }
        }
    }, 100);
}

resetBtn.addEventListener("click", randomWord);
typingInput.addEventListener("input", initGame);
inputs.addEventListener("click", () => typingInput.focus());
document.addEventListener("keydown", () => typingInput.focus());
```

words.js

```
let wordList = [
    {
        word: "python",
        hint: "programming language"
    },
    {
        word: "guitar",
        hint: "a musical instrument"
    },
    {
        word: "aim",
        hint: "a purpose or intention"
```

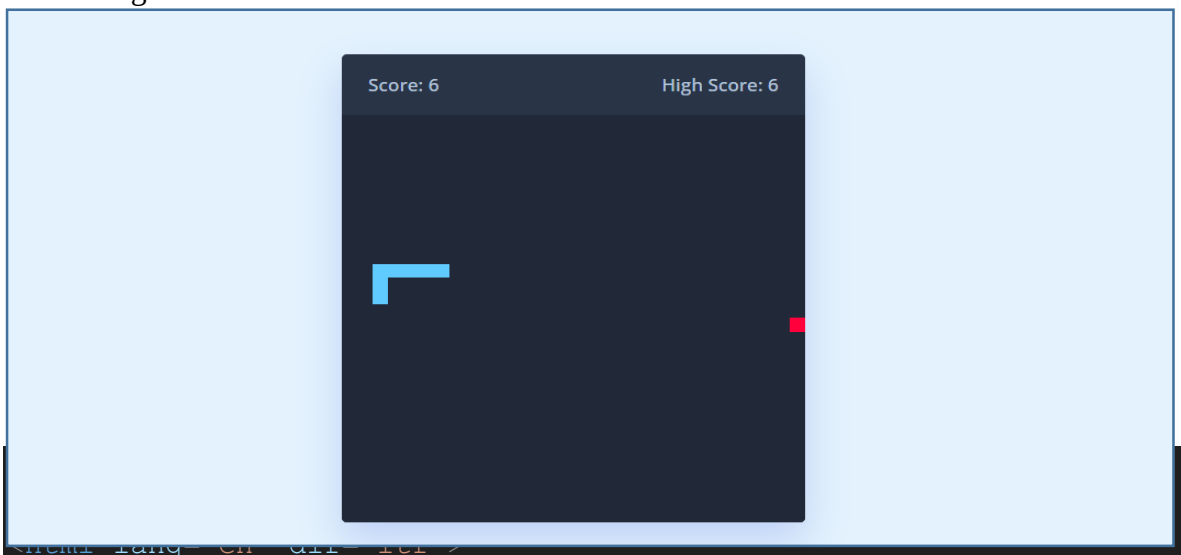
```
    },
    {
        word: "venus",
        hint: "planet of our solar system"
    },
    {
        word: "gold",
        hint: "a yellow precious metal"
    },
    {
        word: "ebay",
        hint: "online shopping site"
    },
    {
        word: "golang",
        hint: "programming language"
    },
    {
        word: "coding",
        hint: "related to programming"
    },
    {
        word: "matrix",
        hint: "science fiction movie"
    },
    {
        word: "bugs",
        hint: "related to programming"
    },
    {
        word: "avatar",
        hint: "epic science fiction film"
    },
    {
        word: "gif",
        hint: "a file format for image"
    },
    {
        word: "mental",
        hint: "related to the mind"
    },
    {
        word: "map",
        hint: "diagram represent of an area"
    },
    {
        word: "island",
        hint: "land surrounded by water"
```

```
    },
    {
        word: "hockey",
        hint: "a famous outdoor game"
    },
    {
        word: "chess",
        hint: "related to a indoor game"
    },
    {
        word: "viber",
        hint: "a social media app"
    },
    {
        word: "github",
        hint: "code hosting platform"
    },
    {
        word: "png",
        hint: "a image file format"
    },
    {
        word: "silver",
        hint: "precious greyish-white metal"
    },
    {
        word: "mobile",
        hint: "an electronic device"
    },
    {
        word: "gpu",
        hint: "computer component"
    },
    {
        word: "java",
        hint: "programming language"
    },
    {
        word: "google",
        hint: "famous search engine"
    },
    {
        word: "venice",
        hint: "famous city of waters"
    },
    {
        word: "excel",
        hint: "microsoft product for windows"
```

```
    },
    {
        word: "mysql",
        hint: "a relational database system"
    },
    {
        word: "nepal",
        hint: "developing country name"
    },
    {
        word: "flute",
        hint: "a musical instrument"
    },
    {
        word: "crypto",
        hint: "related to cryptocurrency"
    },
    {
        word: "tesla",
        hint: "unit of magnetic flux density"
    },
    {
        word: "mars",
        hint: "planet of our solar system"
    },
    {
        word: "proxy",
        hint: "related to server application"
    },
    {
        word: "email",
        hint: "related to exchanging message"
    },
    {
        word: "html",
        hint: "markup language for the web"
    },
    {
        word: "air",
        hint: "related to a gas"
    },
    {
        word: "idea",
        hint: "a thought or suggestion"
    },
    {
        word: "server",
        hint: "related to computer or system"
```

```
    },
    {
        word: "svg",
        hint: "a vector image format"
    },
    {
        word: "jpeg",
        hint: "a image file format"
    },
    {
        word: "search",
        hint: "act to find something"
    },
    {
        word: "key",
        hint: "small piece of metal"
    },
    {
        word: "egypt",
        hint: "a country name"
    },
    {
        word: "joker",
        hint: "psychological thriller film"
    },
    {
        word: "dubai",
        hint: "developed country name"
    },
    {
        word: "photo",
        hint: "representation of person or scene"
    },
    {
        word: "nile",
        hint: "largest river in the world"
    },
    {
        word: "rain",
        hint: "related to a water"
    },
]
```

3. Snake game




```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Snake Game JavaScript | CodingNepal</title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.3.0/css/all.min.css">
    <script src="script.js" defer></script>
  </head>
  <body>
    <div class="wrapper">
      <div class="game-details">
        <span class="score">Score: 0</span>
        <span class="high-score">High Score: 0</span>
      </div>
      <div class="play-board"></div>
      <div class="controls">
        <i data-key="ArrowLeft" class="fa-solid fa-arrow-left-
long"></i>
        <i data-key="ArrowUp" class="fa-solid fa-arrow-up-long"></i>
        <i data-key="ArrowRight" class="fa-solid fa-arrow-right-
long"></i>
        <i data-key="ArrowDown" class="fa-solid fa-arrow-down-
long"></i>
      </div>
    </div>

  </body>
</html>
```

Style.css

```
/* Import Google font */
```

```css
@import
url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;500
;600;700&display=swap');
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Open Sans', sans-serif;
}
body {
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  background: #E3F2FD;
}
.wrapper {
  width: 65vmin;
  height: 70vmin;
  display: flex;
  overflow: hidden;
  flex-direction: column;
  justify-content: center;
  border-radius: 5px;
  background: #293447;
  box-shadow: 0 20px 40px rgba(52, 87, 220, 0.2);
}
.game-details {
  color: #B8C6DC;
  font-weight: 500;
  font-size: 1.2rem;
  padding: 20px 27px;
  display: flex;
  justify-content: space-between;
}
.play-board {
  height: 100%;
  width: 100%;
  display: grid;
  background: #212837;
  grid-template: repeat(30, 1fr) / repeat(30, 1fr);
}
.play-board .food {
  background: #FF003D;
}
.play-board .head {
  background: #60CBFF;
}
```

```css
.controls {
  display: none;
  justify-content: space-between;
}
.controls i {
  padding: 25px 0;
  text-align: center;
  font-size: 1.3rem;
  color: #B8C6DC;
  width: calc(100% / 4);
  cursor: pointer;
  border-right: 1px solid #171B26;
}

@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;
  }
}
```

Script.js

```js
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;

// Getting high score from the local storage
```

```
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("Game Over! Press OK to replay...");
    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) {
        velocityX = 1;
        velocityY = 0;
    }
}

// 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;
        }
    }
    playBoard.innerHTML = html;
}

updateFoodPosition();
setIntervalId = setInterval(initGame, 100);
document.addEventListener("keyup", changeDirection);
```