Source code example to encrypt string values coding example

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
  <style>
    .btn{
      padding:5px;
      font-size:0.9em;
      display:inline-block;
    }
    .inputer{
      display:block;
    }
  </style>
</head>
<body>
  <div class="output">

  </div>

  <script src='app22.js'></script>
</body>
</html>
const output = document.querySelector('.output');
const myInput = adder(output,false,'input','inputer');
const btn1 = adder(output,'Encrypt','button','btn');
const btn2 = adder(output,'Decrypt','button','btn');
const message = adder(output,'','div','message');
myInput.value = "Laurence Svekis";

btn1.addEventListener('click',encrypto);
btn2.addEventListener('click',decrypto);

function encrypto(){
    let val = btoa(myInput.value);
    myInput.value = val;
    console.log(val);
    message.textContent= val;
}
```

```
function decrypto(){
    let val = atob(myInput.value);
    myInput.value = val;
    console.log(val);
    message.textContent= val;
}



function adder(parent,html,eleT,cla){
    const ele = document.createElement(eleT);
    if(html) ele.innerHTML = html;
    if(cla) ele.classList.add(cla);
    return parent.appendChild(ele);
}
```

Source Code JavaScript Roulette Game with CSS grid Dynamic page Grid

```
<!DOCTYPE html>
<html>

<head>
  <title>JavaScript Game</title>
  <style>
    * {
      box-sizing: border-box;
    }

    .bet {
      position: absolute;
      max-width: 25px;
      padding: 3px;
      left: 0;
      right: 0;
      margin-left: auto;
      margin-right: auto;
      background-color: yellow;
      overflow: hidden;
```

```
      border-radius: 50%;
      color: black;
    }

    .output {
      display: grid;
      width: 80vw;
      height: 80vh;
      margin: auto;
      padding: 0;
    }

    .box {
      position: relative;
      text-align: center;
      border: 1px solid #ddd;
    }
  </style>
</head>

<body>
  <div class="gamearea">
  </div>
  <script src="game1.js"></script>
</body>

</html>
const gamearea = document.querySelector('.gamearea');
const score = createEle(gamearea, 'div', 'Score :', 'score');
const btn = createEle(gamearea, 'button', 'Spin', 'btn');
const message = createEle(gamearea, 'div', 'Press Spin', 'message');
const output = createEle(gamearea, 'div', '', 'output');
const game = {
    x: 7,
    y: 9,
    coins: 50,
    sel: [],
    eles: [],
    winner: false,
    styler: ['black', 'white']
};
const total = game.x * game.y;
btn.disabled = true;
btn.addEventListener('click', spinner);
```

```
createBoard();
updateScore();

function spinner() {
    btn.disabled = true;
    const ranVal = Math.floor(Math.random() * total) + 1;
    console.log(ranVal);
    game.winner = ranVal - 1;
    game.styler = [game.eles[ranVal - 1].style.backgroundColor, game.eles[ranVal -
1].style.color];
    const win = game.sel.includes(ranVal);
    console.log(win);

    const eles = output.querySelectorAll('.bet');
    eles.forEach((el) => {
        el.remove();
        console.log(el);
    })
    if (win) {
        const winAmount = total;
        game.coins += winAmount;
        message.innerHTML = `Winner on ${ranVal} you won ${winAmount}`;
        createEle(game.eles[ranVal - 1], 'div', '$', 'bet');
        game.eles[ranVal - 1].style.backgroundColor = 'green';
    } else {
        message.innerHTML = `Lost sorry you did not bet on ${ranVal}`;
        game.eles[ranVal - 1].style.backgroundColor = 'purple';
    }
    game.sel = [];
    updateScore();

    game.eles.forEach((el) => {
        el.bet = false;
    })

}

function createBoard() {
    for (let i = 0; i < total; i++) {
        const temp = createEle(output, 'div', `${i+1}`, 'box');
        if (i % 2) {
            temp.style.backgroundColor = 'red';
        } else {
```

```
            temp.style.backgroundColor = 'black';
            temp.style.color = 'white';
        }
        game.eles.push(temp);
        temp.bet = false;
        temp.addEventListener('click', (e) => {
            btn.disabled = false;
            if (game.winner) {
                const parTemp = game.eles[game.winner];
                parTemp.style.backgroundColor = game.styler[0];
                parTemp.style.color = game.styler[1];
                game.winner = false;
                const bets = parTemp.querySelector('.bet');
                if (bets) {
                    bets.remove();
                }
            }
            console.log(temp.textContent);
            if (temp.bet) {
                console.log(game.winner);
                const bets = temp.querySelector('.bet');
                bets.remove();
                //console.log(bets);
                temp.bet = false;
                game.coins++;
                const index = game.sel.indexOf(i + 1);
                if (index > -1) {
                    game.sel.splice(index, 1);
                }
            } else {
                game.sel.push(i + 1);
                game.coins--;
                temp.bet = true;
                createEle(temp, 'div', '$', 'bet');
            }
            updateScore();
        }, true);
    }
    output.style.setProperty(`grid-template-columns`, `repeat(${game.x},1fr)`)
}

function updateScore() {
    score.innerHTML = `Coins : ${game.coins}`;
    console.log(game.sel);
```

```
}

function createEle(parent, eleType, html, eleClass) {
    const ele = document.createElement(eleType);
    ele.innerHTML = html;
    ele.classList.add(eleClass);
    return parent.appendChild(ele);
}

/*
const div = document.createElement('div');
div.innerHTML  = 'Hello World';
div.classList.add('myClass');
gamearea.append(div);
*/
```