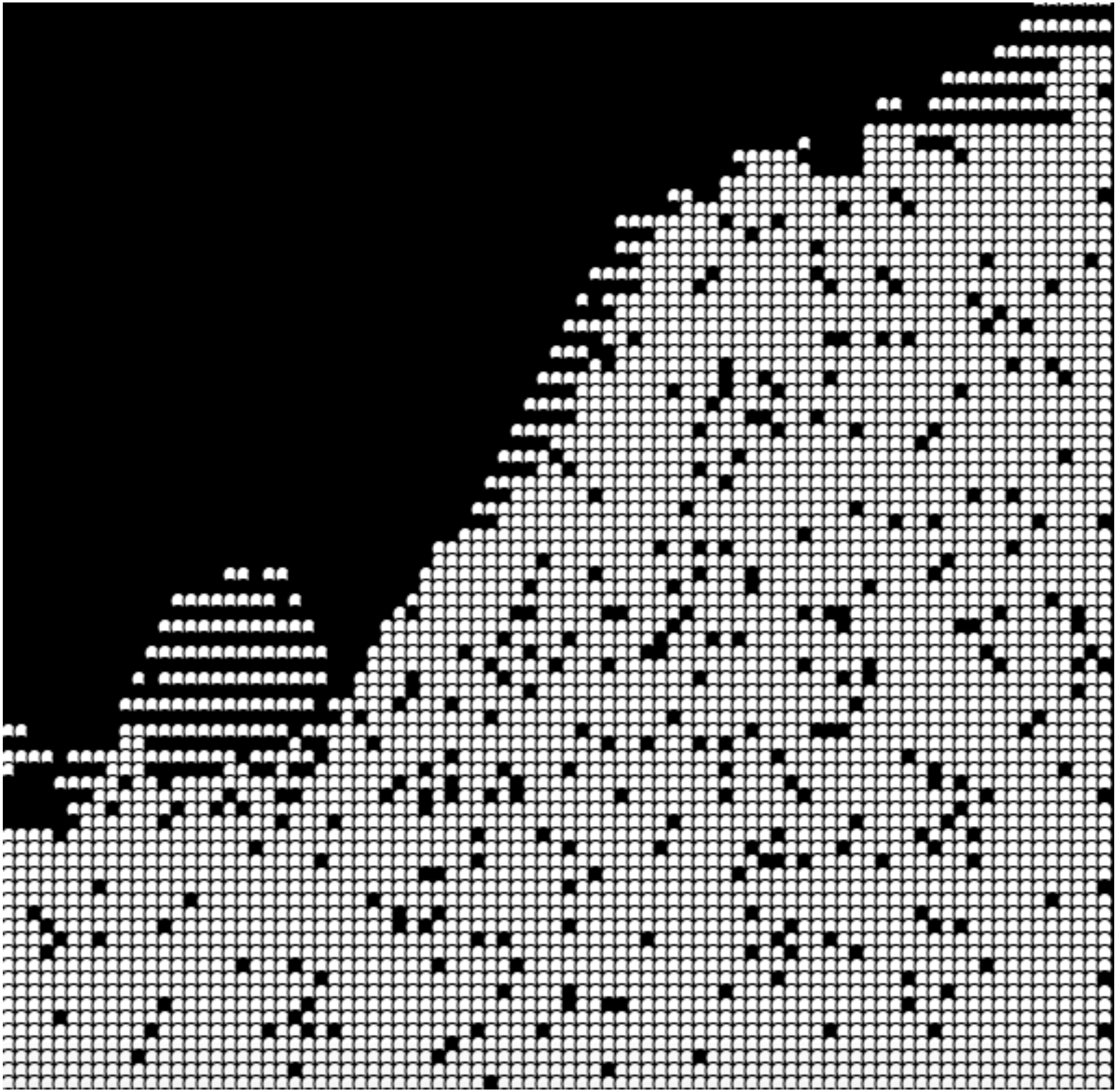
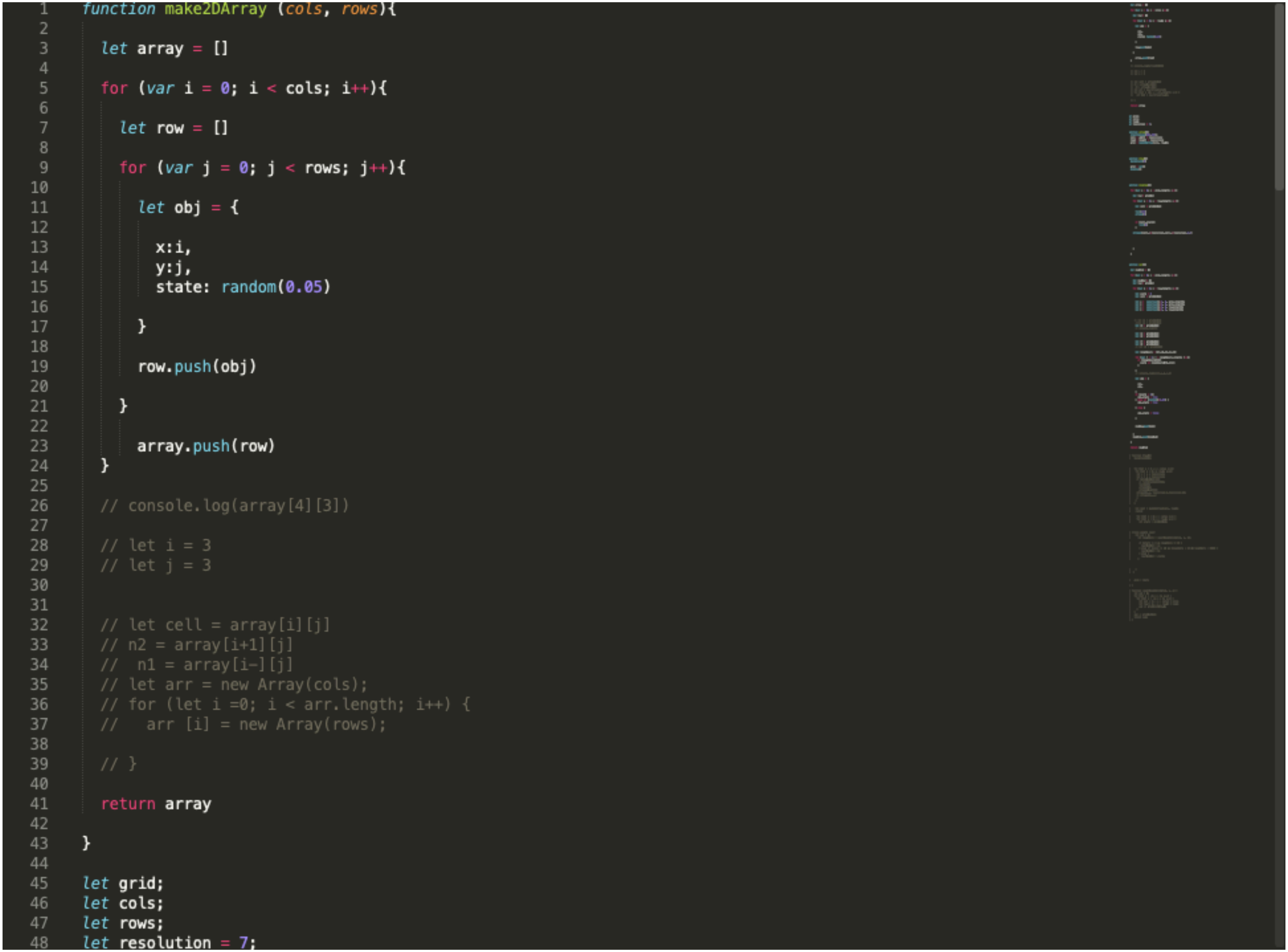
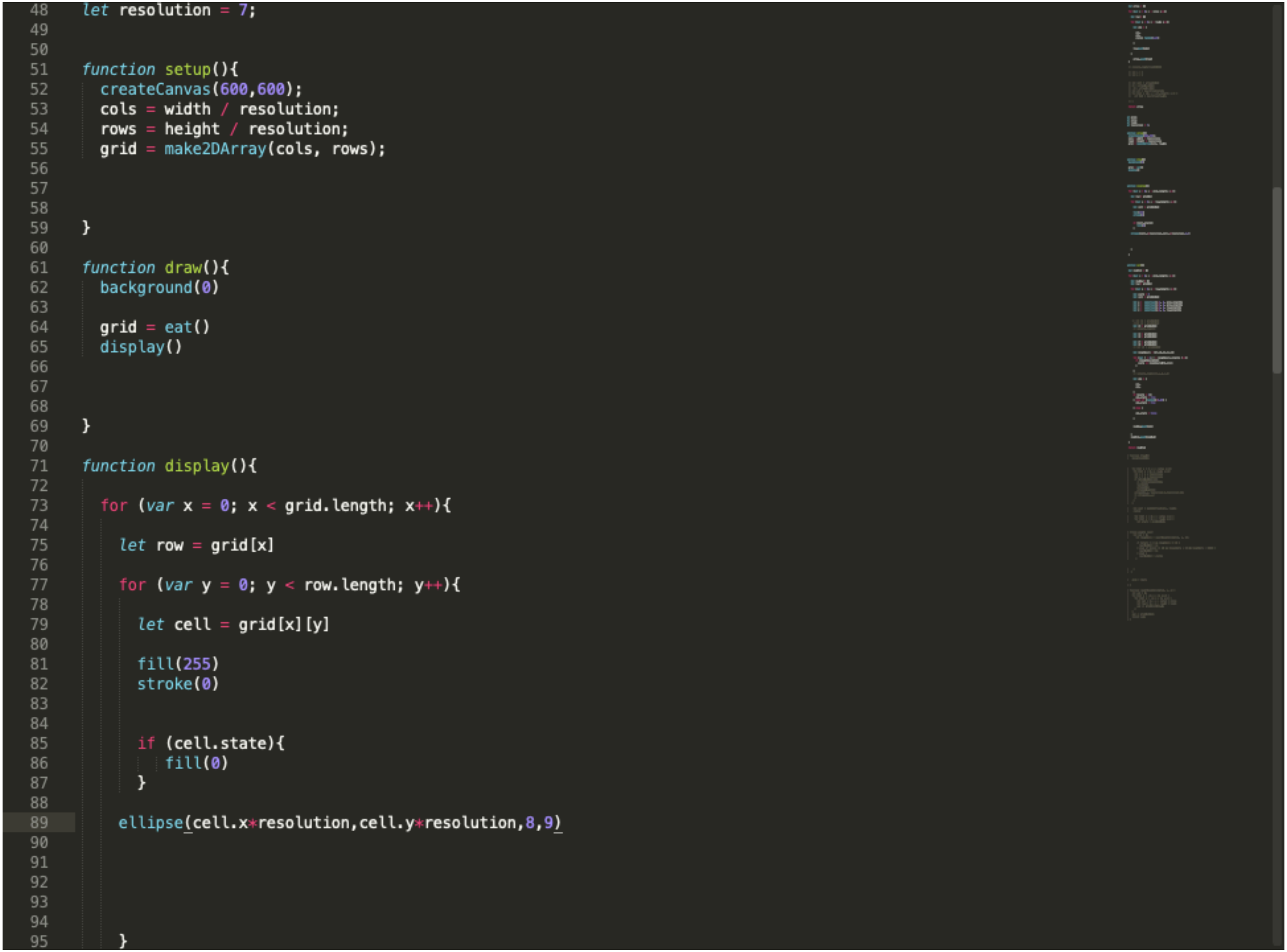
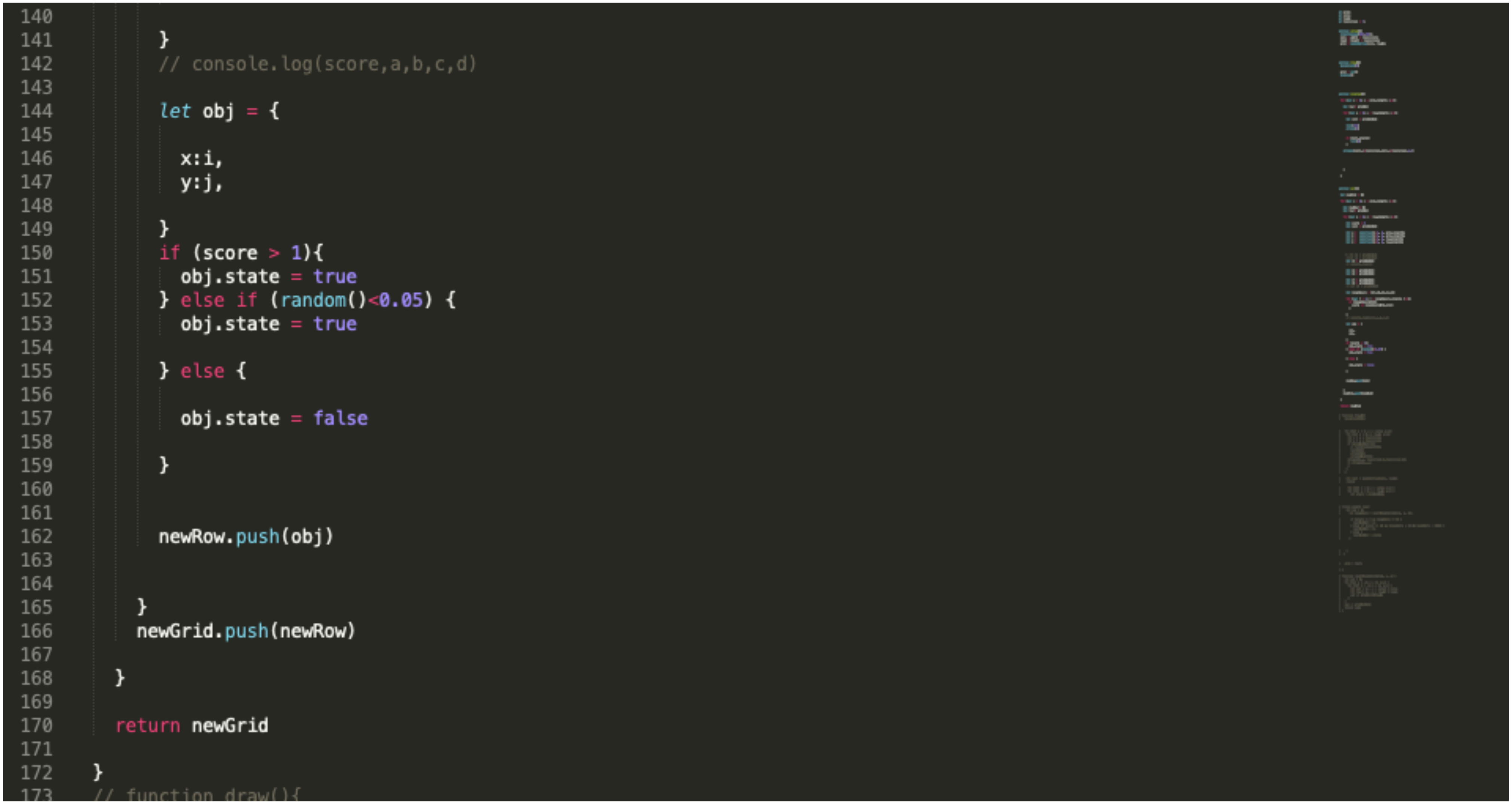




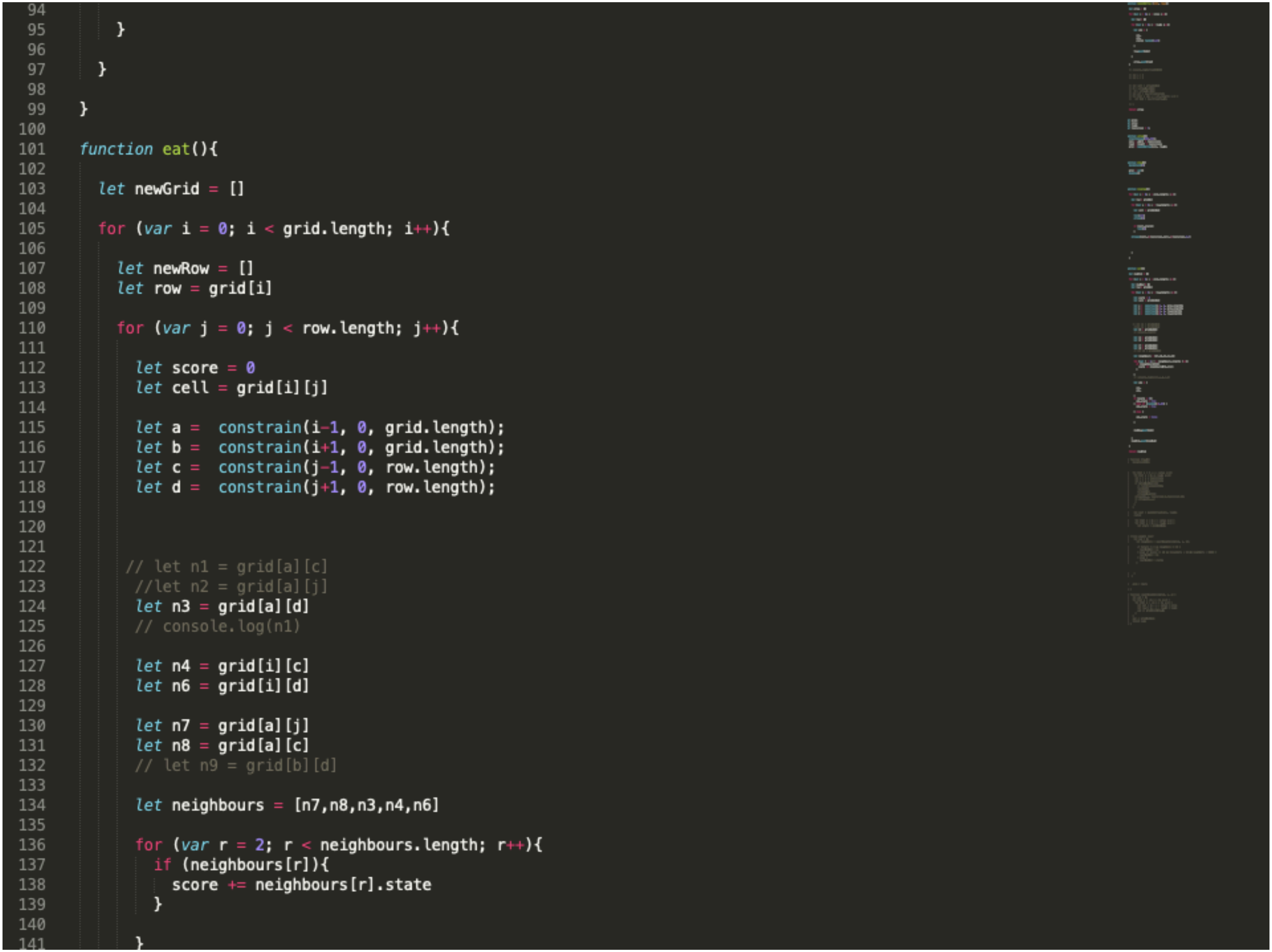
```
function make2DArray (cols, rows){

  let array = []

  for (var i = 0; i < cols; i++){

    let row = []

    for (var j = 0; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: random(0.05)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 7;
```



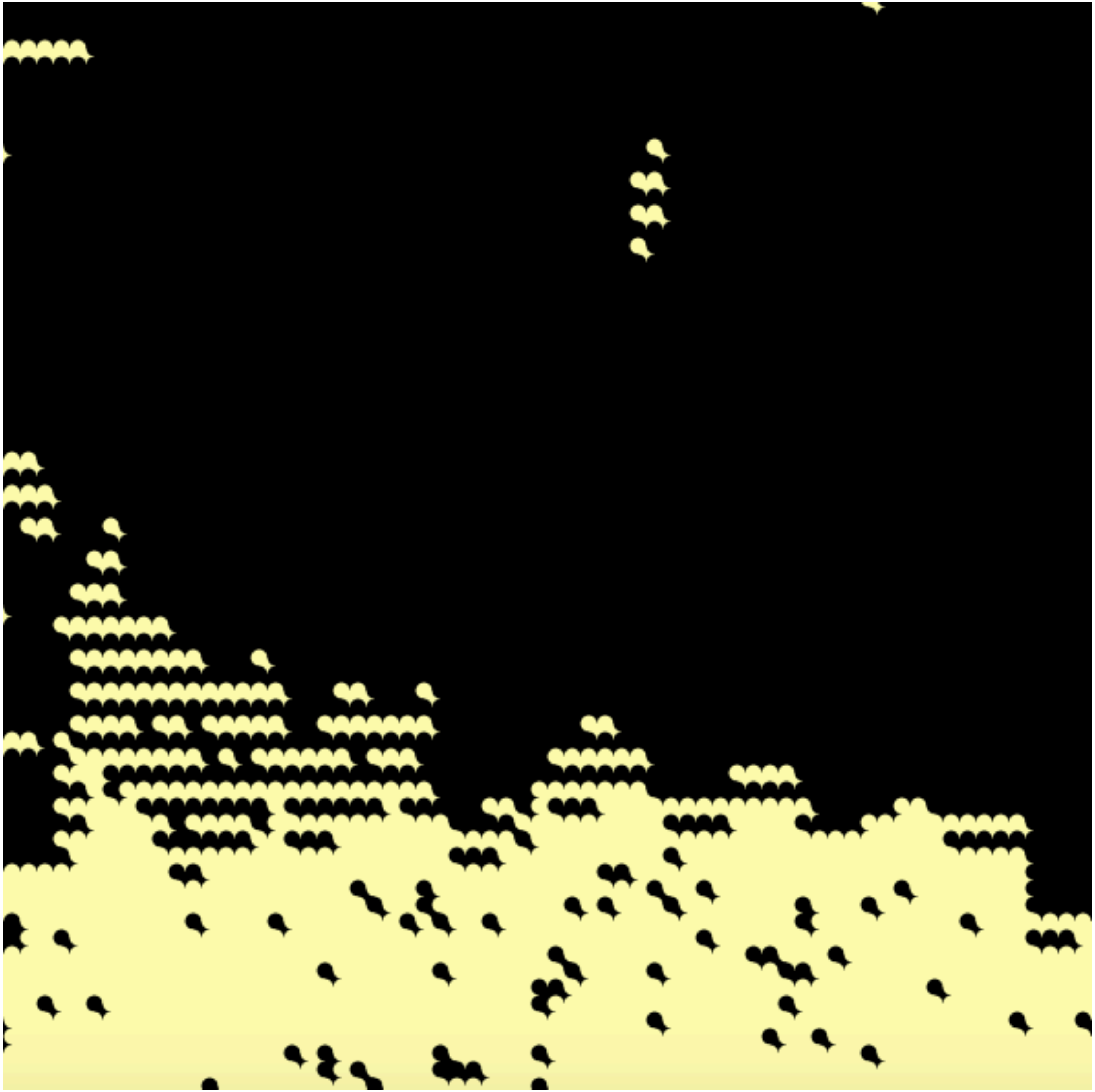
```
let resolution = 7;


function setup(){
  createCanvas(600,600);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols, rows);



}

function draw(){
  background(0)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(255)
      stroke(0)


      if (cell.state){
        fill(0)
      }

    ellipse(cell.x*resolution,cell.y*resolution,8,9)






    }
```

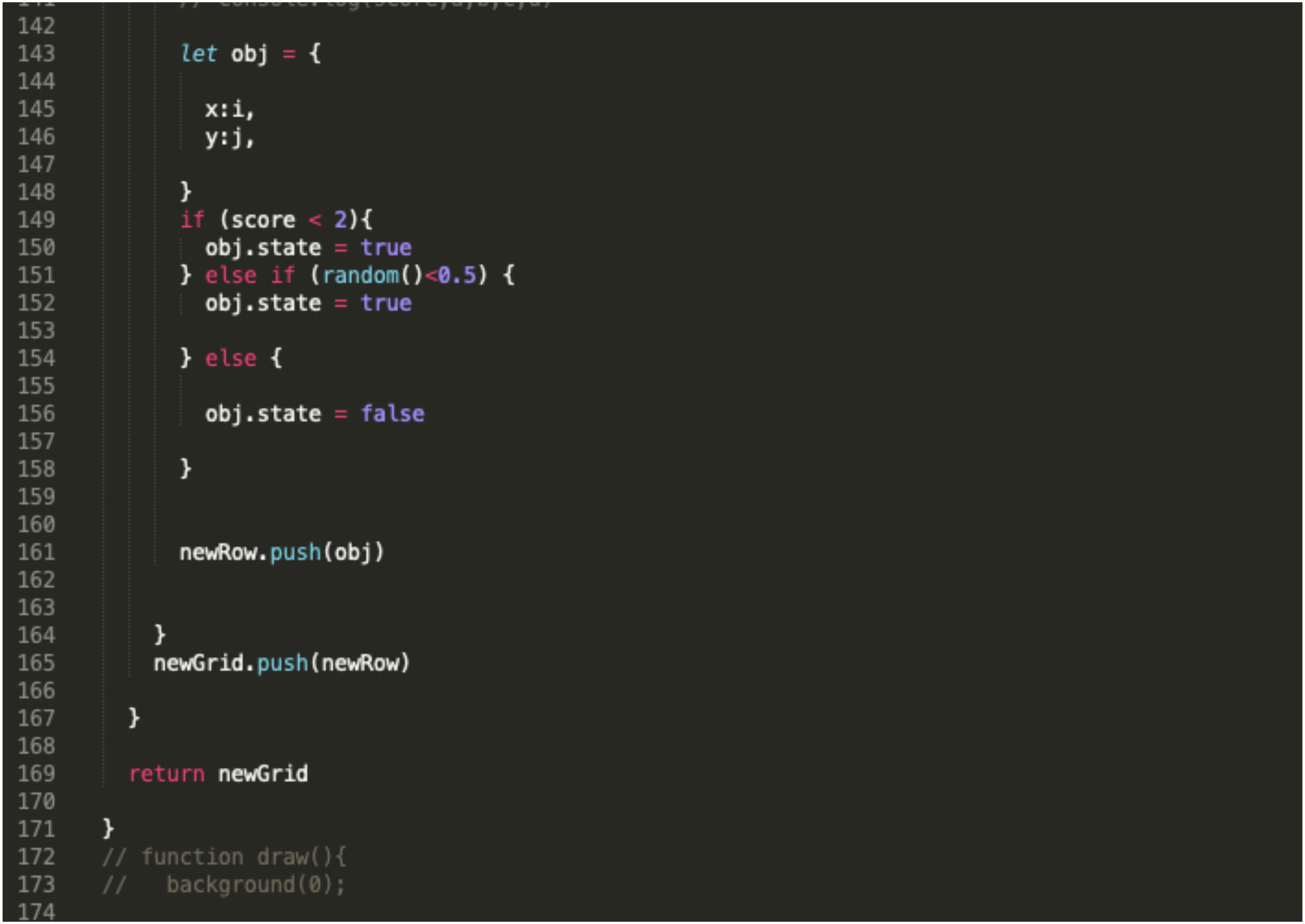


```

      }
      // console.log(score,a,b,c,d)

      let obj = {

        x:i,
        y:j,

      }
      if (score > 1){
        obj.state = true
      } else if (random()<0.05) {
        obj.state = true

      } else {

        obj.state = false

      }


      newRow.push(obj)


    }
    newGrid.push(newRow)

  }

  return newGrid

}
// function draw(){
```

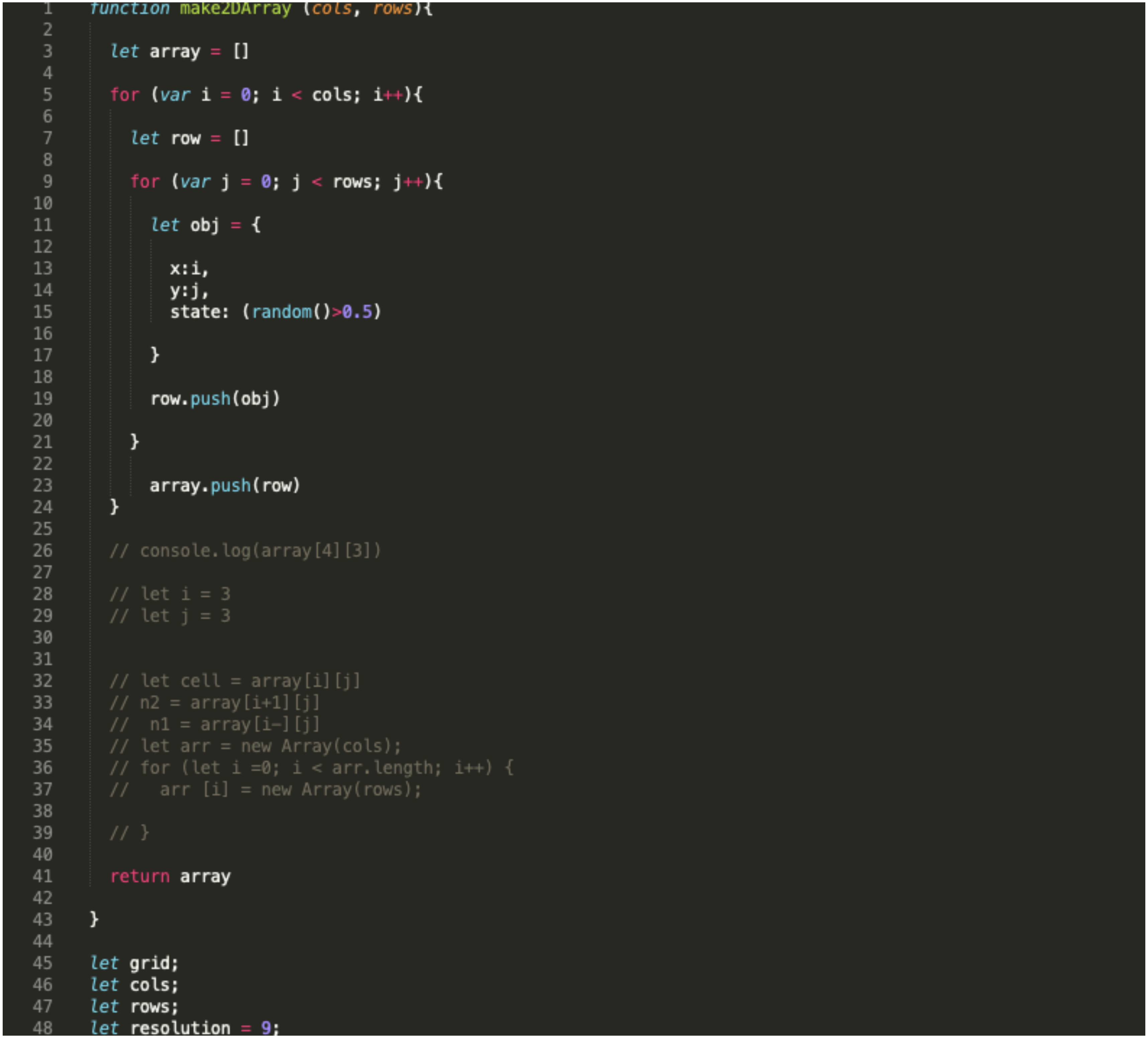


```
    }

  }

}

function eat(){

  let newGrid = []

  for (var i = 0; i < grid.length; i++){

    let newRow = []
    let row = grid[i]

    for (var j = 0; j < row.length; j++){

      let score = 0
      let cell = grid[i][j]

      let a =  constrain(i-1, 0, grid.length);
      let b =  constrain(i+1, 0, grid.length);
      let c =  constrain(j-1, 0, row.length);
      let d =  constrain(j+1, 0, row.length);



     // let n1 = grid[a][c]
      //let n2 = grid[a][j]
      let n3 = grid[a][d]
      // console.log(n1)

      let n4 = grid[i][c]
      let n6 = grid[i][d]

      let n7 = grid[a][j]
      let n8 = grid[a][c]
      // let n9 = grid[b][d]

      let neighbours = [n7,n8,n3,n4,n6]

      for (var r = 2; r < neighbours.length; r++){
        if (neighbours[r]){
          score += neighbours[r].state
        }

      }
```

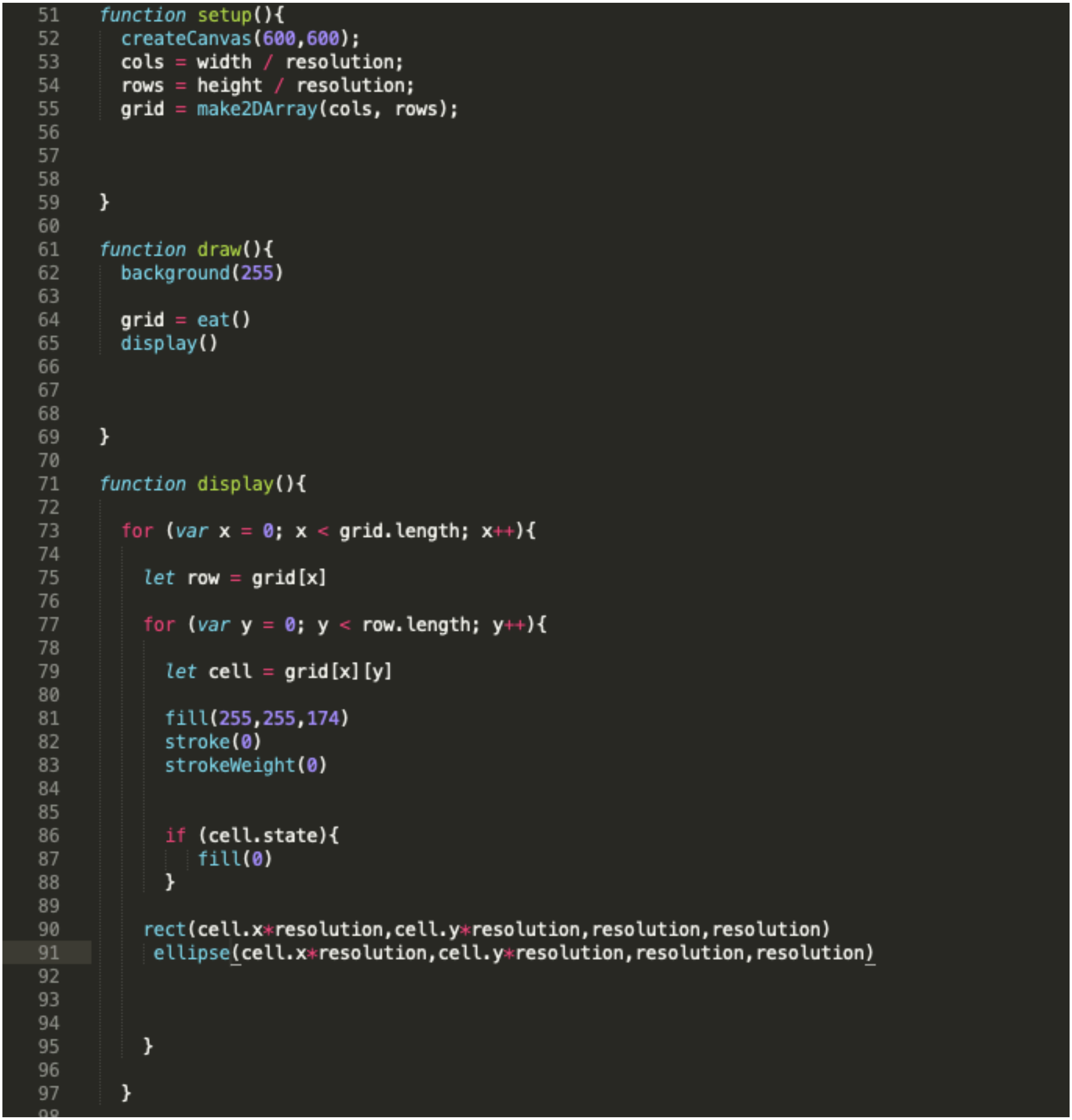




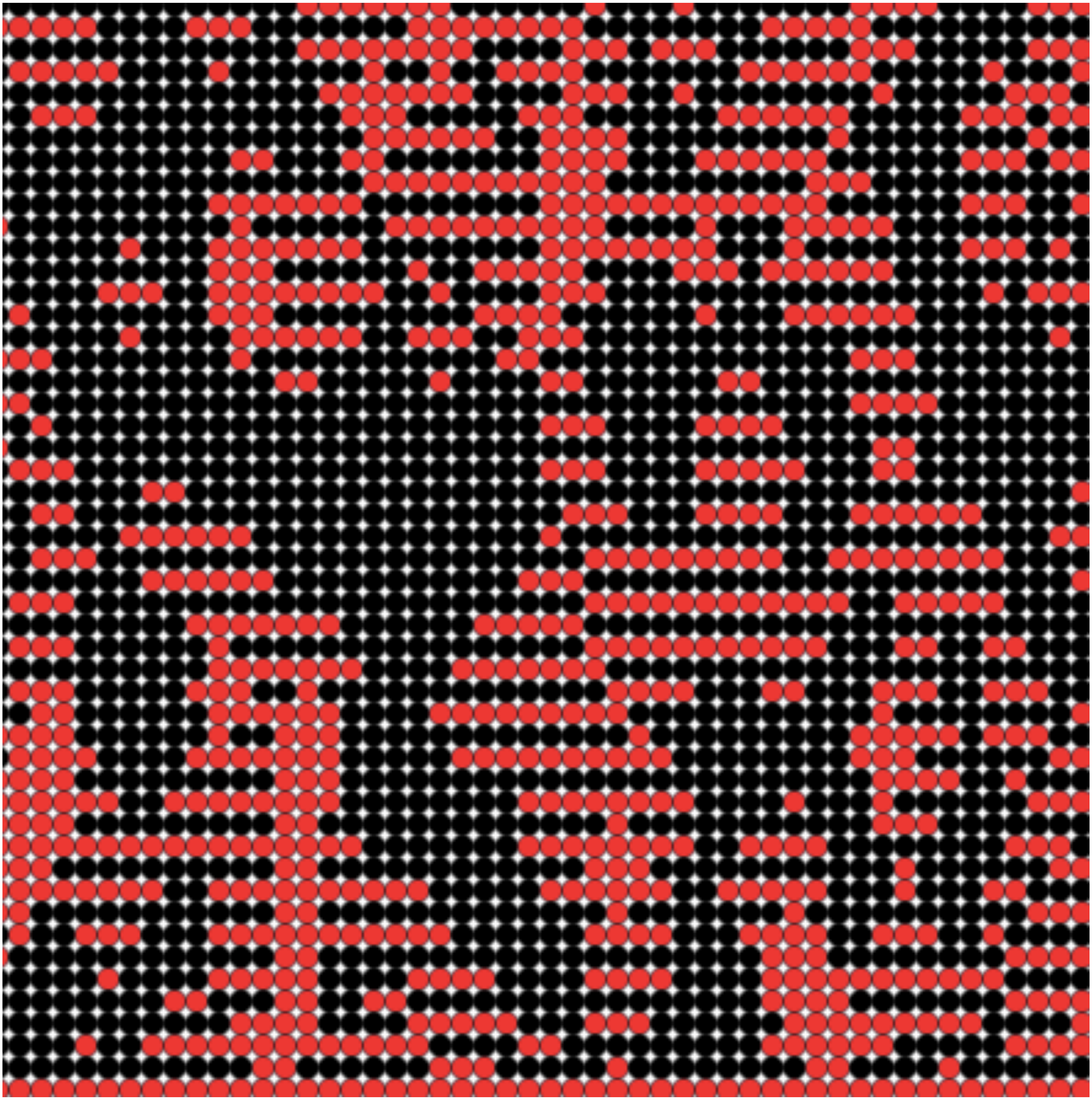
```
      // console.log(score,a,b,c,d)

      let obj = {

        x:i,
        y:j,

      }
      if (score < 2){
        obj.state = true
      } else if (random()<0.5) {
        obj.state = true

      } else {

        obj.state = false

      }


      newRow.push(obj)


    }
    newGrid.push(newRow)

  }

  return newGrid

}
// function draw(){
//   background(0);

```

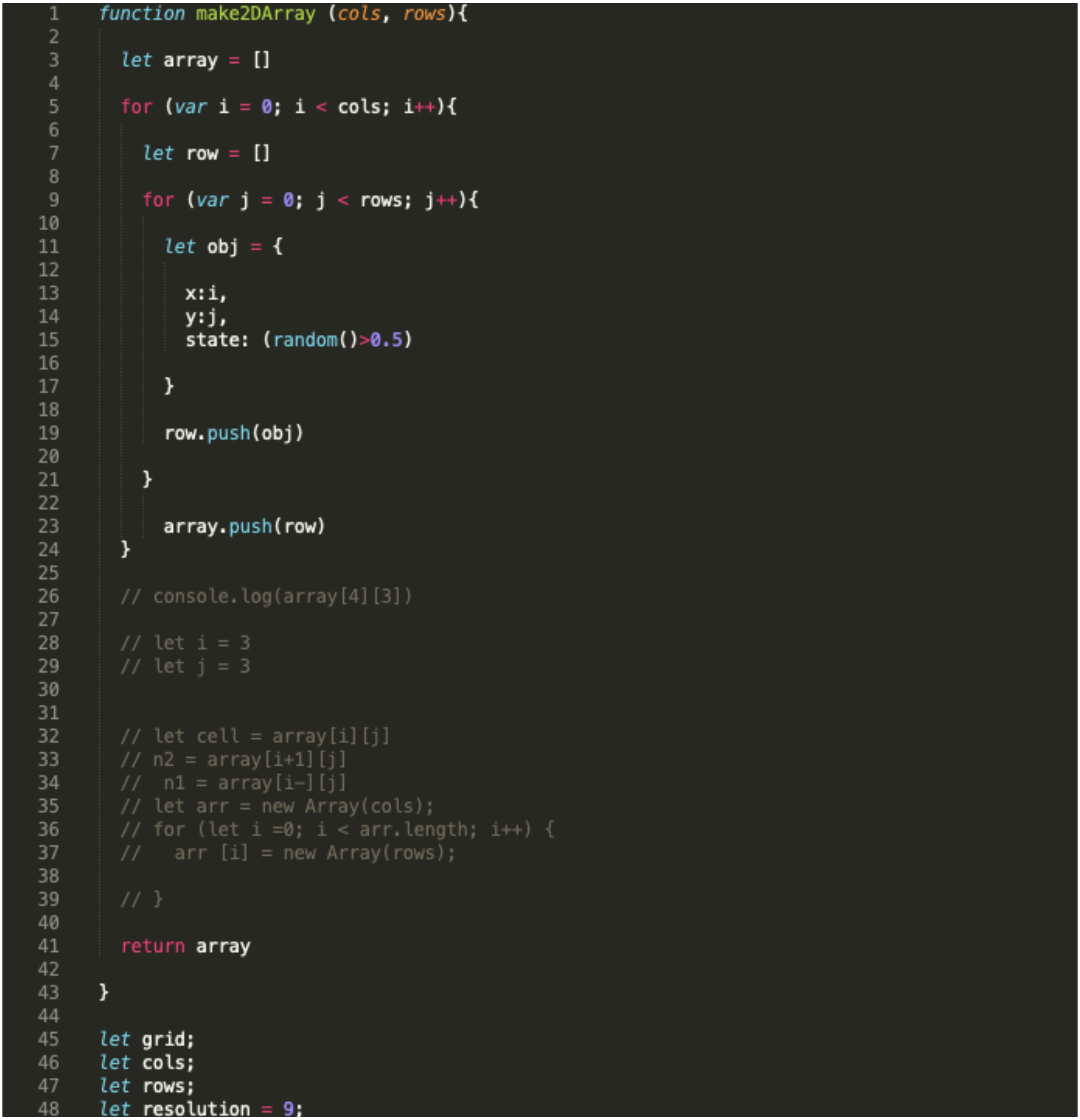


```
function make2DArray (cols, rows){

  let array = []

  for (var i = 0; i < cols; i++){

    let row = []

    for (var j = 0; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: (random()>0.5)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 9;
```

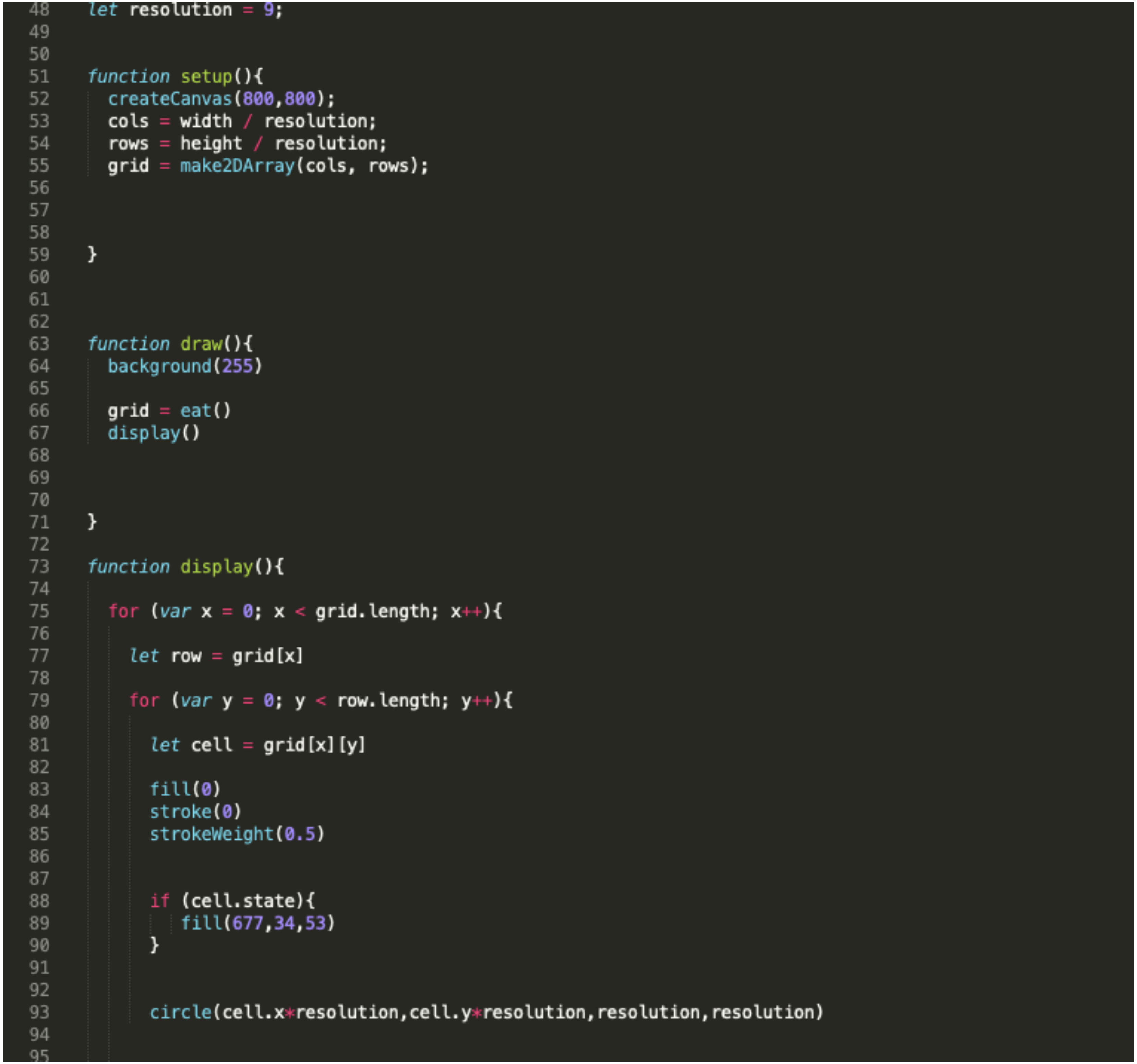


```
function setup(){
  createCanvas(600,600);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols, rows);



}

function draw(){
  background(255)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(255,255,174)
      stroke(0)
      strokeWeight(0)


      if (cell.state){
          fill(0)
      }

    rect(cell.x*resolution,cell.y*resolution,resolution,resolution)
     ellipse(cell.x*resolution,cell.y*resolution,resolution,resolution)



    }

  }
```

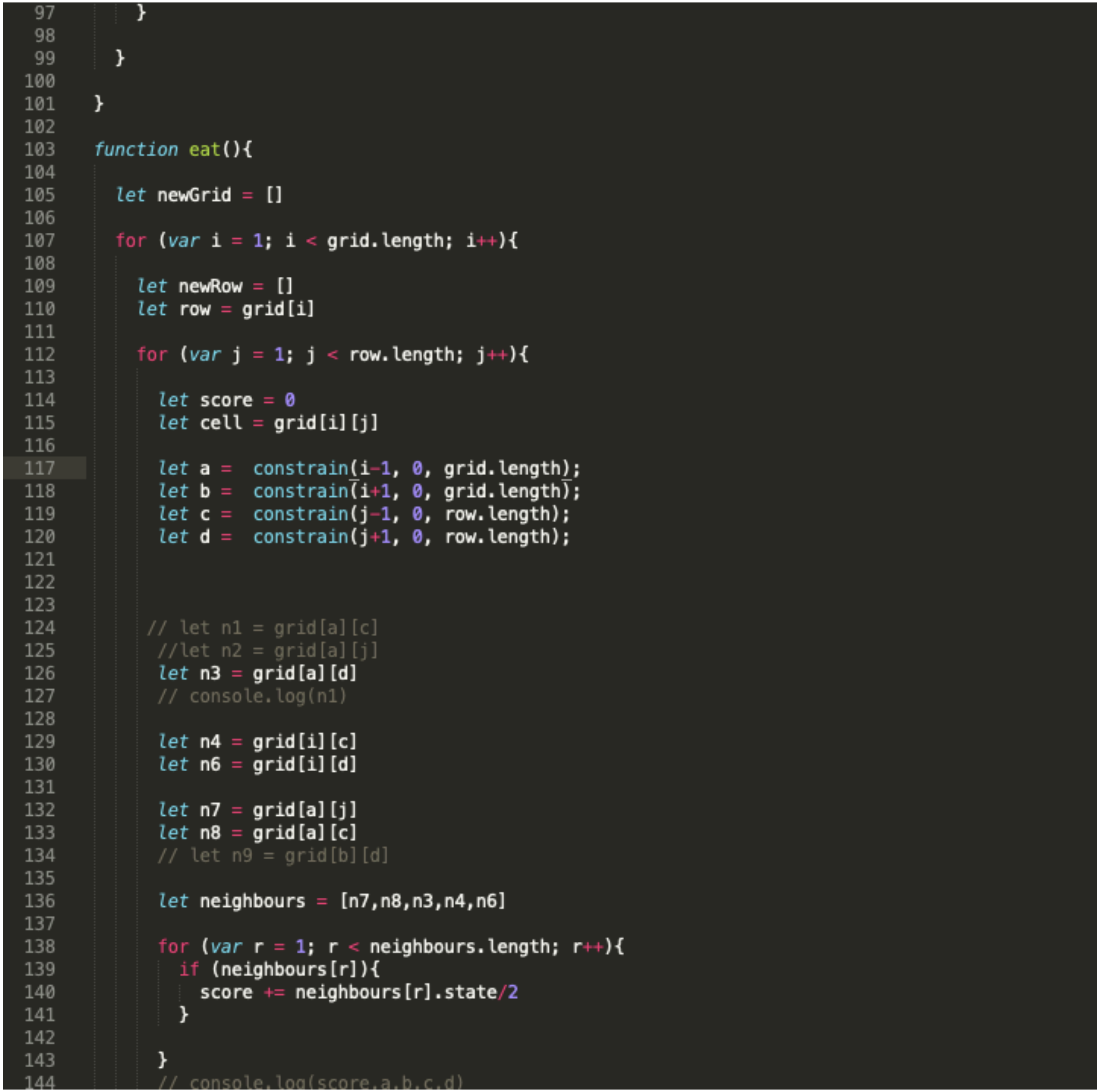




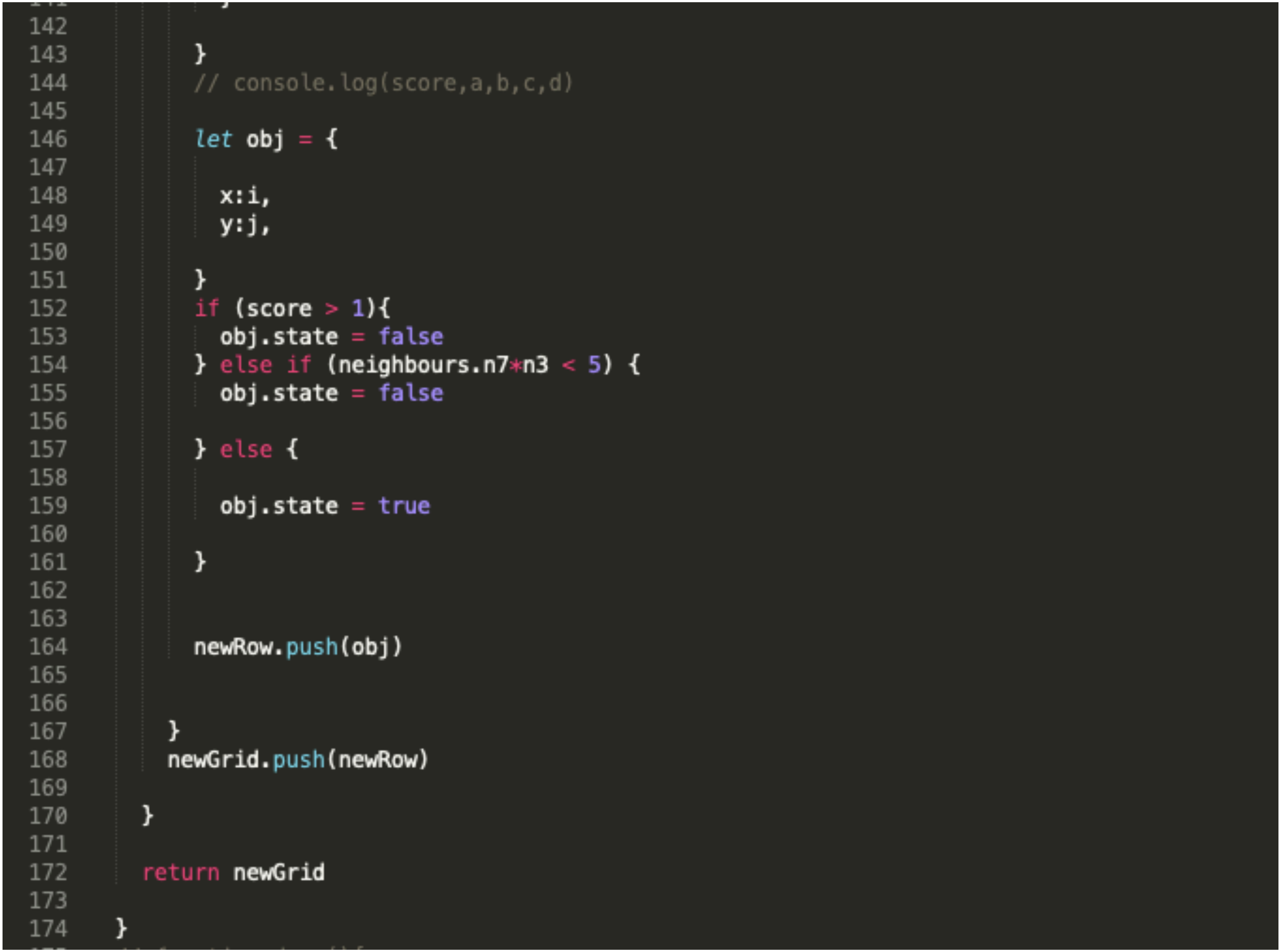
```
function make2DArray (cols, rows){

  let array = []

  for (var i = 0; i < cols; i++){

    let row = []

    for (var j = 0; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: (random()>0.5)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 9;
```



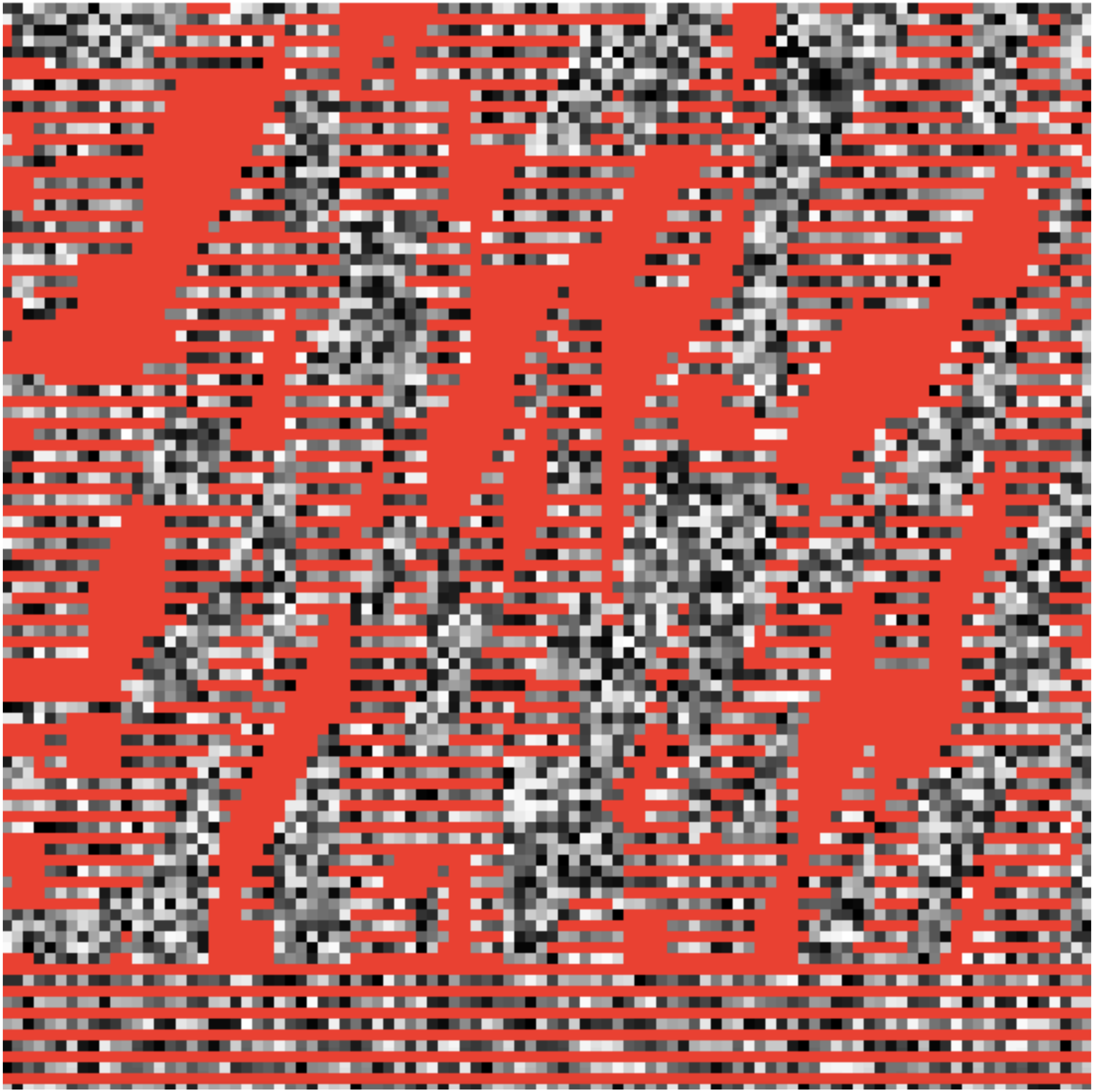
```
let resolution = 9;


function setup(){
  createCanvas(800,800);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols, rows);



}



function draw(){
  background(255)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(0)
      stroke(0)
      strokeWeight(0.5)


      if (cell.state){
        fill(677,34,53)
      }


      circle(cell.x*resolution,cell.y*resolution,resolution,resolution)

```

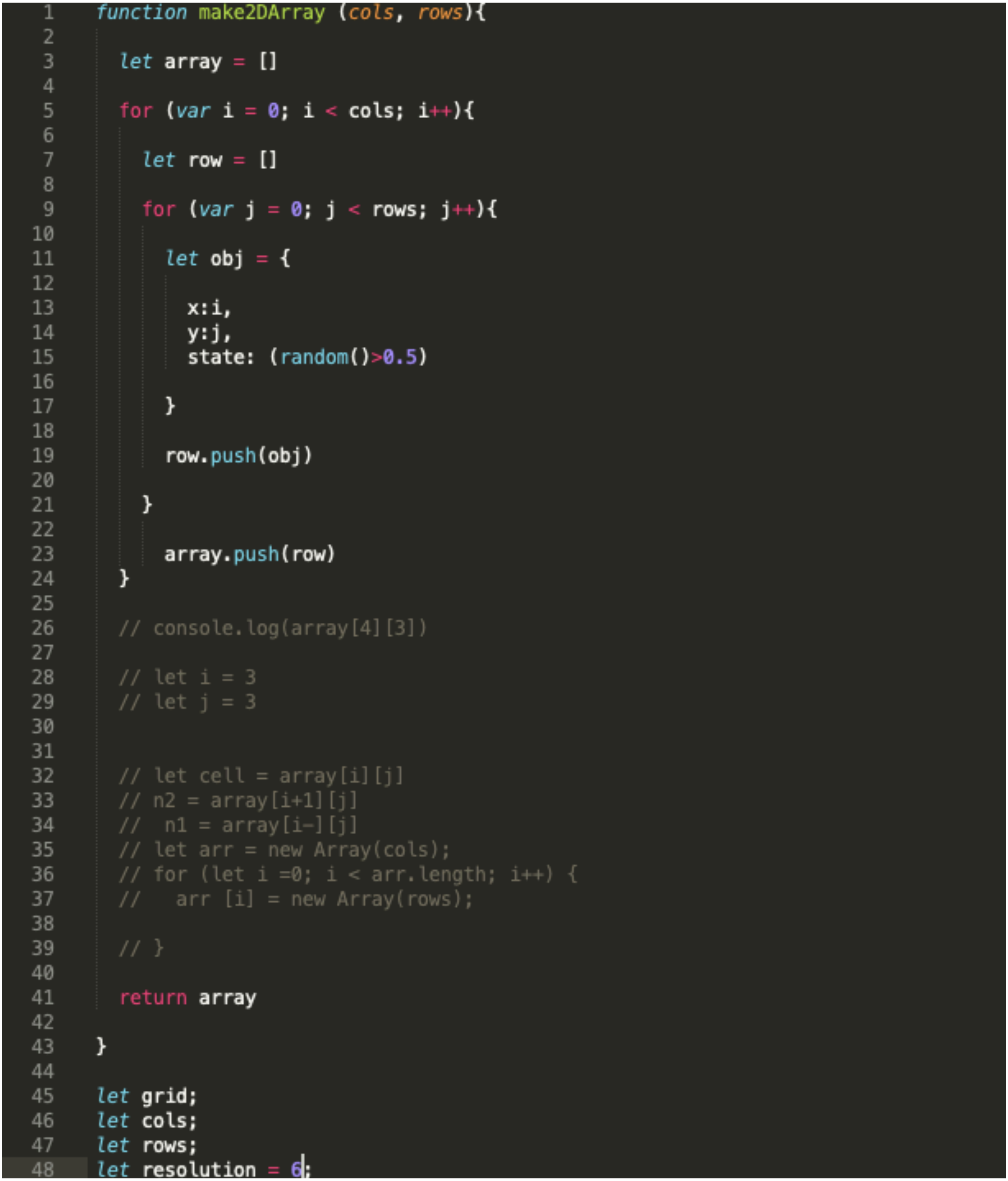


```
    }

  }

}

function eat(){

  let newGrid = []

  for (var i = 1; i < grid.length; i++){

    let newRow = []
    let row = grid[i]

    for (var j = 1; j < row.length; j++){

      let score = 0
      let cell = grid[i][j]

      let a =  constrain(i-1, 0, grid.length);
      let b =  constrain(i+1, 0, grid.length);
      let c =  constrain(j-1, 0, row.length);
      let d =  constrain(j+1, 0, row.length);



     // let n1 = grid[a][c]
      //let n2 = grid[a][j]
      let n3 = grid[a][d]
      // console.log(n1)

      let n4 = grid[i][c]
      let n6 = grid[i][d]

      let n7 = grid[a][j]
      let n8 = grid[a][c]
      // let n9 = grid[b][d]

      let neighbours = [n7,n8,n3,n4,n6]

      for (var r = 1; r < neighbours.length; r++){
        if (neighbours[r]){
          score += neighbours[r].state/2
        }

      }
      // console.log(score,a,b,c,d)
```

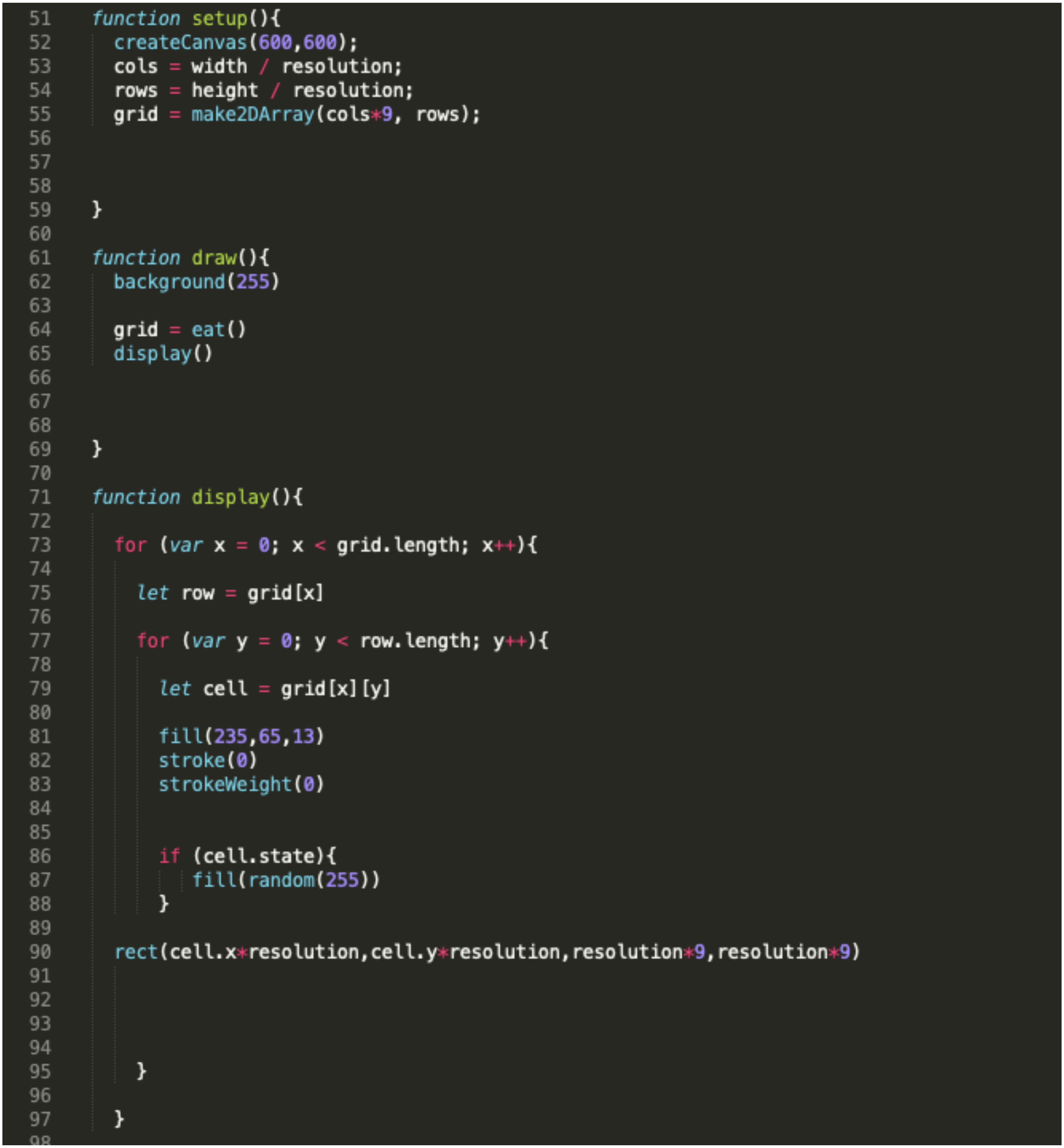


```
   }
   // console.log(score,a,b,c,d)

   let obj = {

    x:i,
    y:j,

   }
   if (score > 1){
    obj.state = false
   } else if (neighbours.n7*n3 < 5) {
    obj.state = false

   } else {

    obj.state = true

   }


   newRow.push(obj)


  }
  newGrid.push(newRow)

 }

 return newGrid

}
```

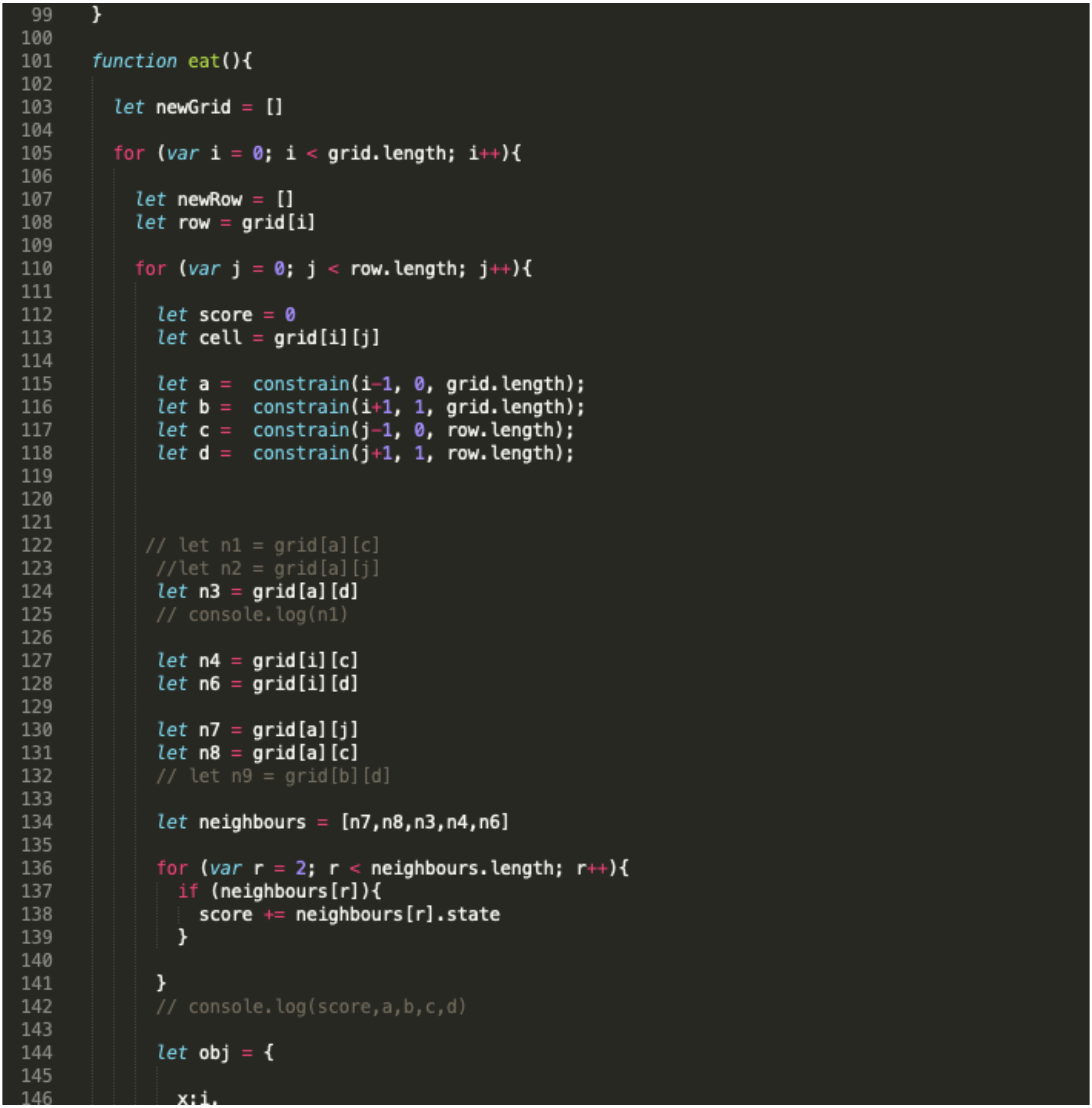




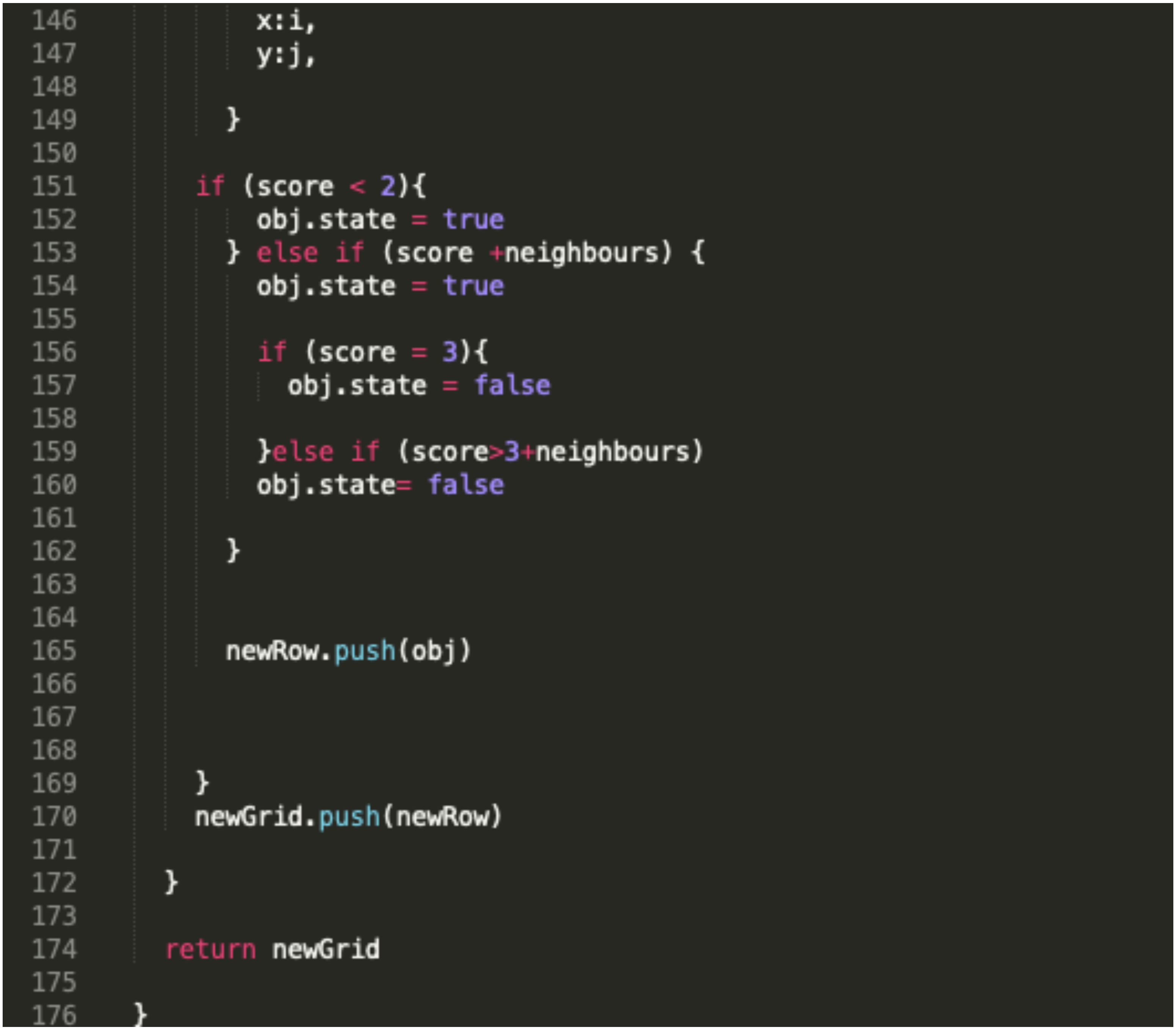
```
function make2DArray (cols, rows){

  let array = []

  for (var i = 0; i < cols; i++){

    let row = []

    for (var j = 0; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: (random()>0.5)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 6;
```



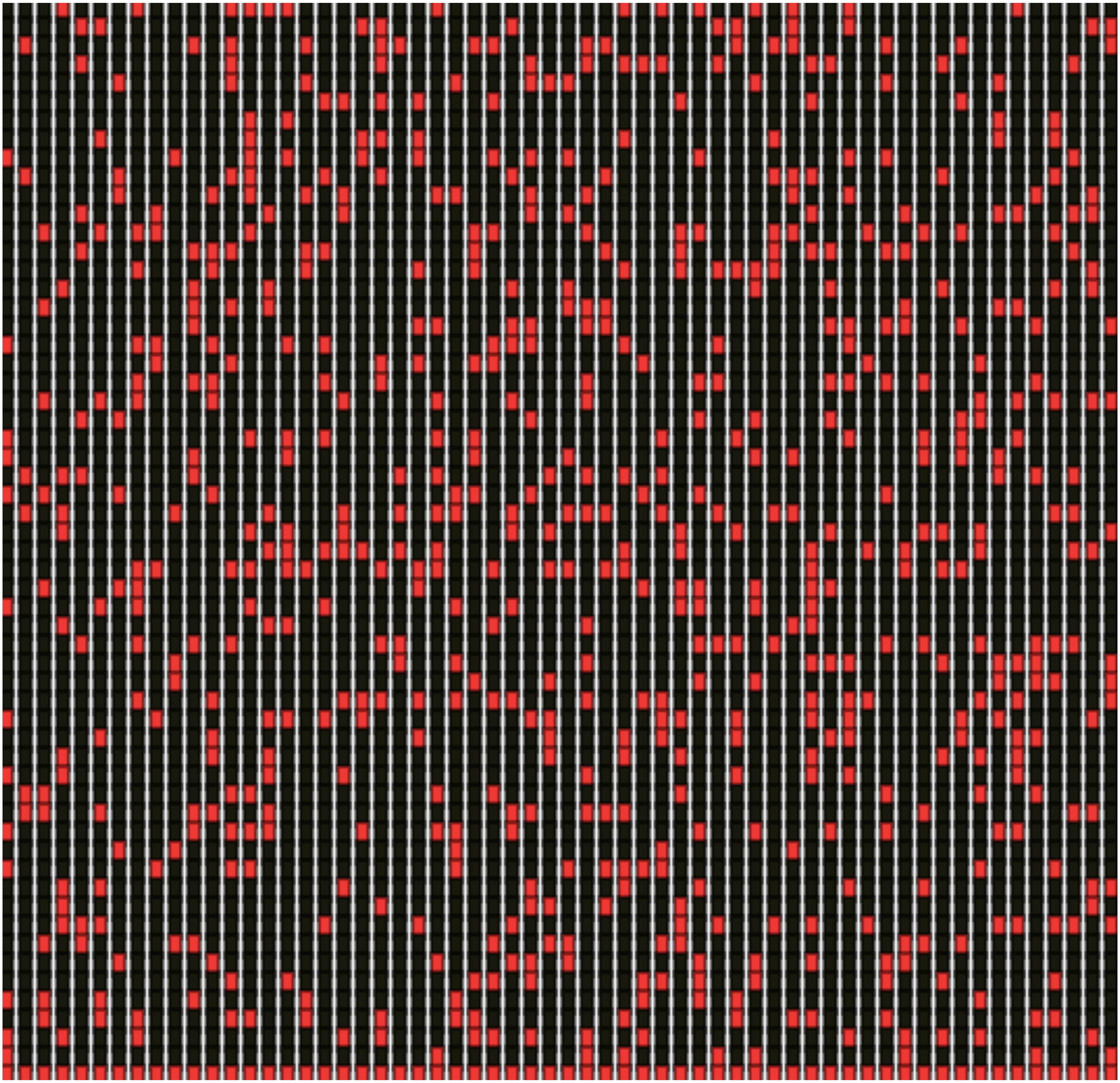
```
function setup(){
  createCanvas(600,600);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols*9, rows);



}

function draw(){
  background(255)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(235,65,13)
      stroke(0)
      strokeWeight(0)


      if (cell.state){
         fill(random(255))
      }

  rect(cell.x*resolution,cell.y*resolution,resolution*9,resolution*9)




    }

  }
```

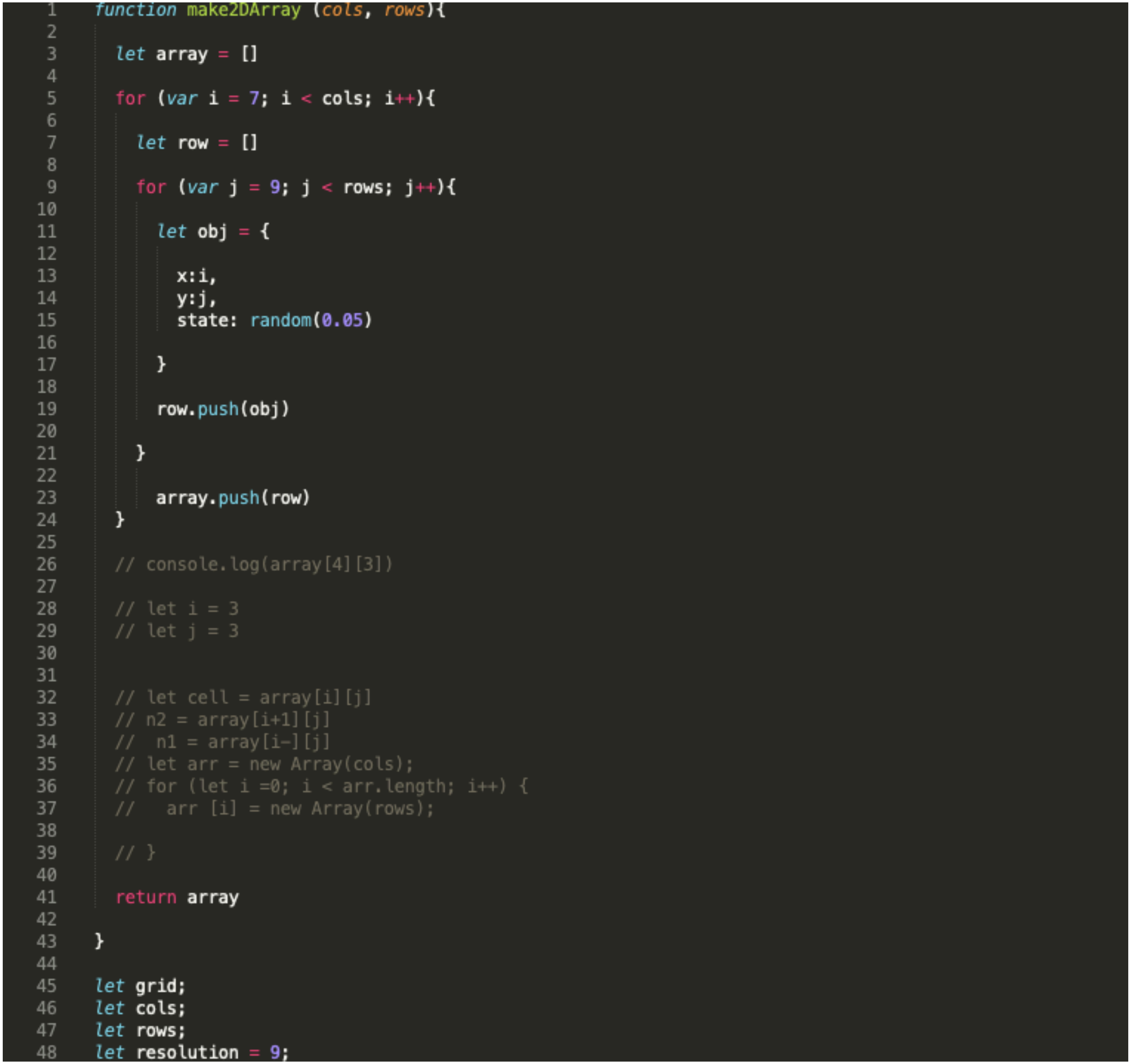


```
}

function eat(){

  let newGrid = []

  for (var i = 0; i < grid.length; i++){

    let newRow = []
    let row = grid[i]

    for (var j = 0; j < row.length; j++){

      let score = 0
      let cell = grid[i][j]

      let a =  constrain(i-1, 0, grid.length);
      let b =  constrain(i+1, 1, grid.length);
      let c =  constrain(j-1, 0, row.length);
      let d =  constrain(j+1, 1, row.length);



     // let n1 = grid[a][c]
      //let n2 = grid[a][j]
      let n3 = grid[a][d]
      // console.log(n1)

      let n4 = grid[i][c]
      let n6 = grid[i][d]

      let n7 = grid[a][j]
      let n8 = grid[a][c]
      // let n9 = grid[b][d]

      let neighbours = [n7,n8,n3,n4,n6]

      for (var r = 2; r < neighbours.length; r++){
        if (neighbours[r]){
          score += neighbours[r].state
        }

      }
      // console.log(score,a,b,c,d)

      let obj = {

        x:i,
```

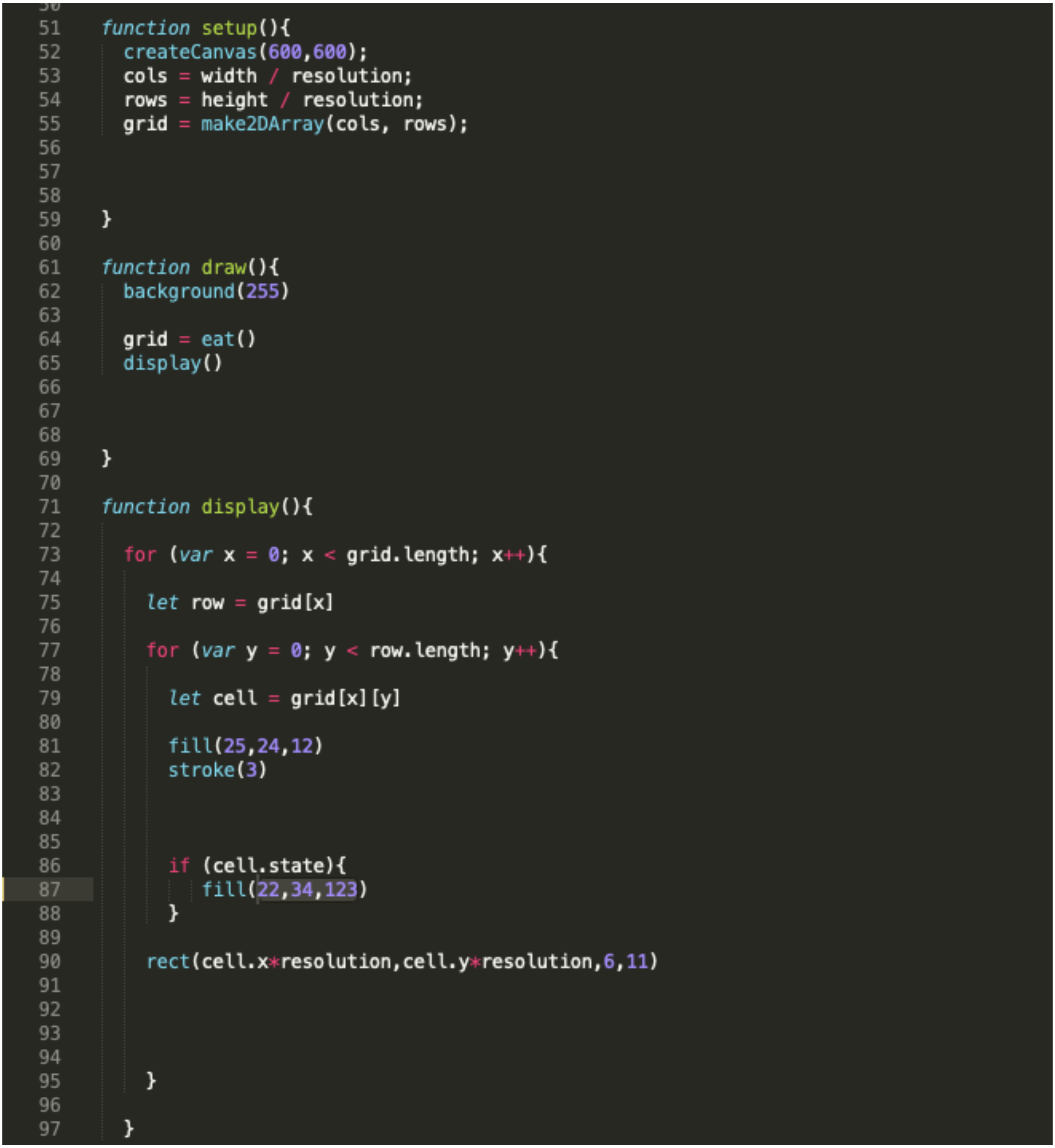


```
        x:i,
        y:j,

      }

    if (score < 2){
        obj.state = true
      } else if (score +neighbours) {
        obj.state = true

        if (score = 3){
          obj.state = false

        }else if (score>3+neighbours)
        obj.state= false

      }


      newRow.push(obj)



    }
    newGrid.push(newRow)

  }

  return newGrid

}
```

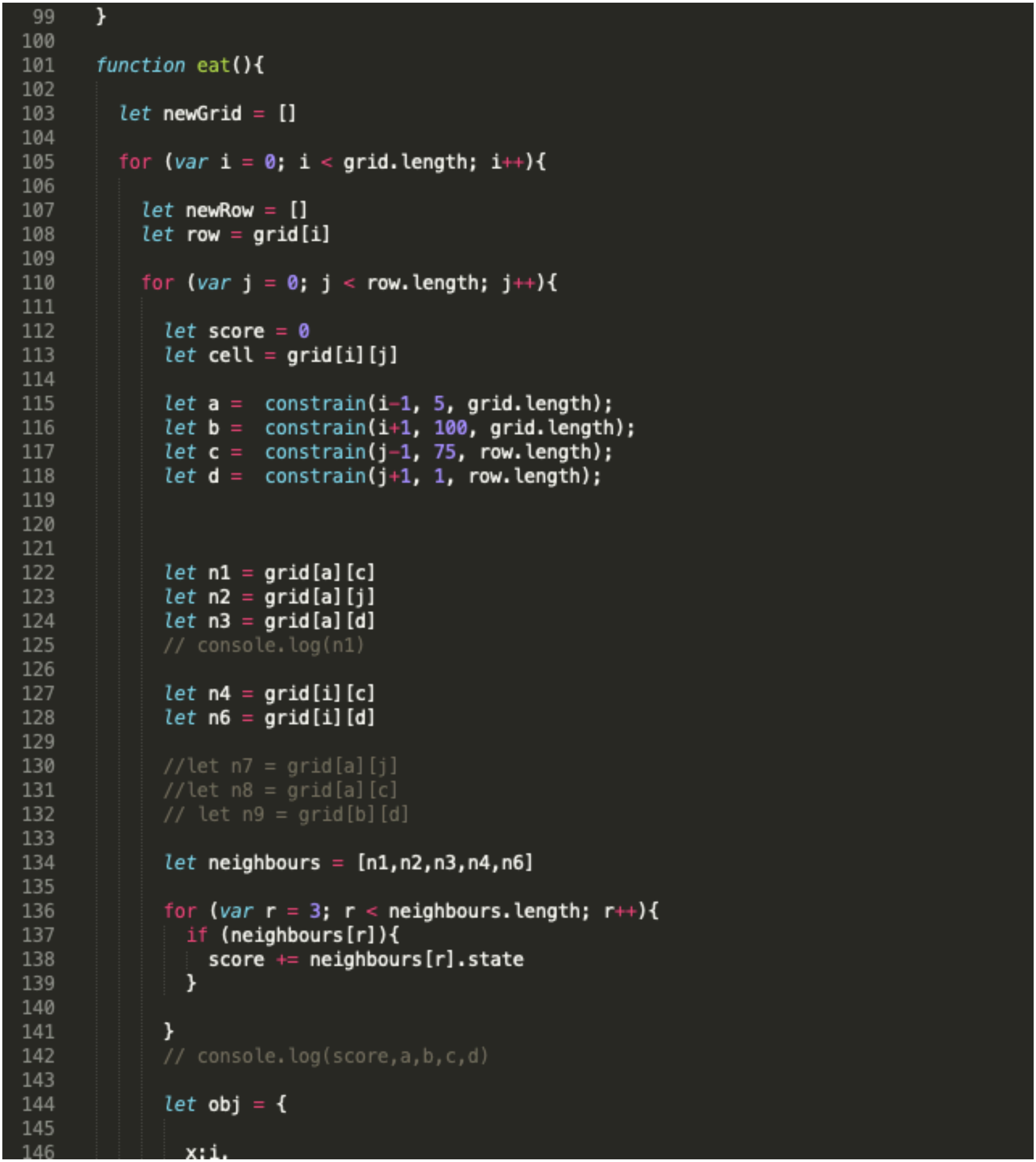




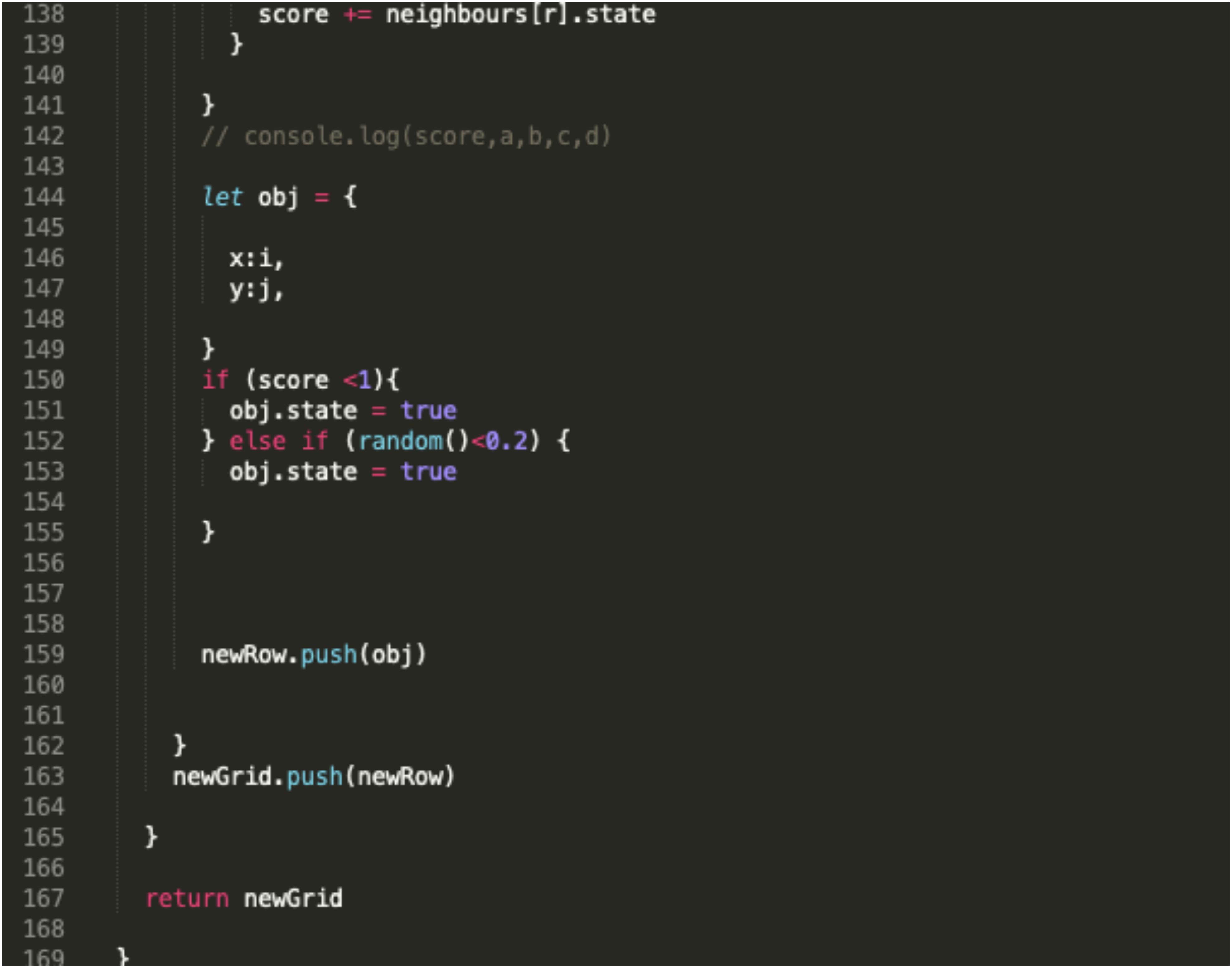
```
function make2DArray (cols, rows){

  let array = []

  for (var i = 7; i < cols; i++){

    let row = []

    for (var j = 9; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: random(0.05)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 9;
```



```
function setup(){
  createCanvas(600,600);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols, rows);



}

function draw(){
  background(255)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(25,24,12)
      stroke(3)



      if (cell.state){
        fill(22,34,123)
      }

    rect(cell.x*resolution,cell.y*resolution,6,11)




    }

  }
```



```
}

function eat(){

  let newGrid = []

  for (var i = 0; i < grid.length; i++){

    let newRow = []
    let row = grid[i]

    for (var j = 0; j < row.length; j++){

      let score = 0
      let cell = grid[i][j]

      let a =  constrain(i-1, 5, grid.length);
      let b =  constrain(i+1, 100, grid.length);
      let c =  constrain(j-1, 75, row.length);
      let d =  constrain(j+1, 1, row.length);



      let n1 = grid[a][c]
      let n2 = grid[a][j]
      let n3 = grid[a][d]
      // console.log(n1)

      let n4 = grid[i][c]
      let n6 = grid[i][d]

      //let n7 = grid[a][j]
      //let n8 = grid[a][c]
      // let n9 = grid[b][d]

      let neighbours = [n1,n2,n3,n4,n6]

      for (var r = 3; r < neighbours.length; r++){
        if (neighbours[r]){
          score += neighbours[r].state
        }

      }
      // console.log(score,a,b,c,d)

      let obj = {

        x:i,
```



```
          score += neighbours[r].state
        }

      }
      // console.log(score,a,b,c,d)

      let obj = {

        x:i,
        y:j,

      }
      if (score <1){
        obj.state = true
      } else if (random()<0.2) {
        obj.state = true

      }



      newRow.push(obj)


    }
    newGrid.push(newRow)

  }

  return newGrid

}
```