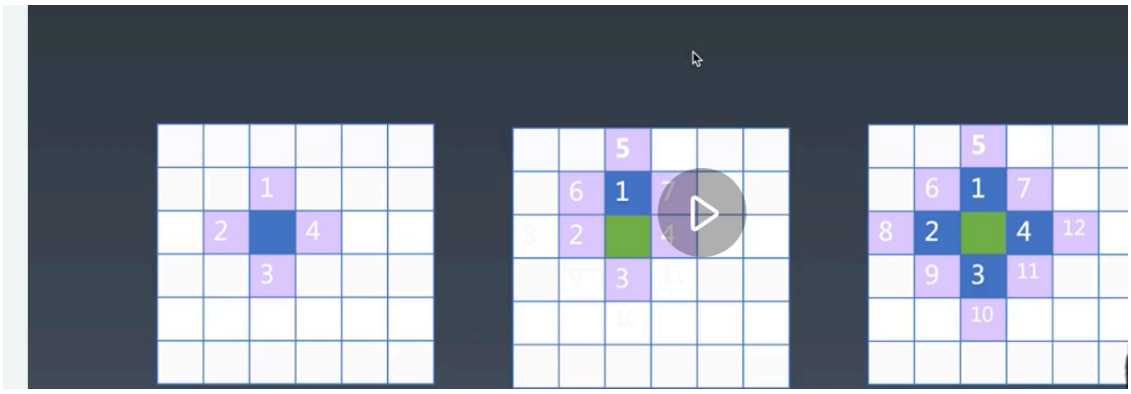
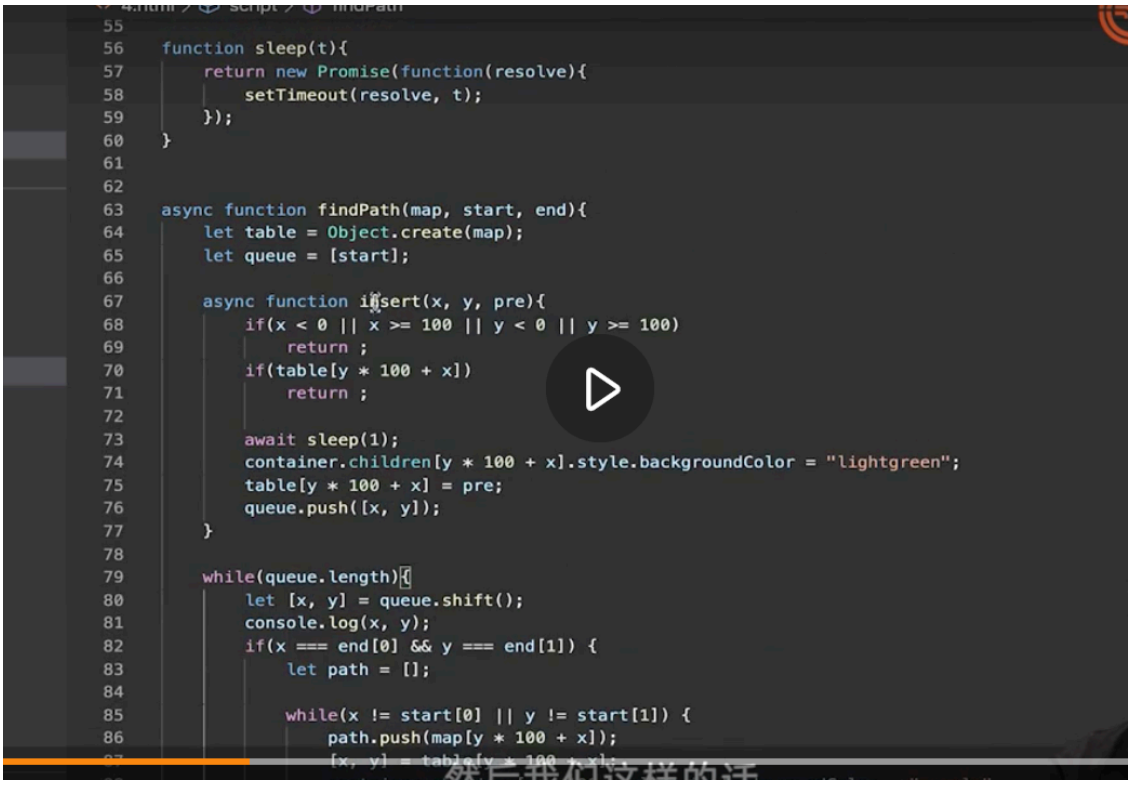



```
54
55
56  function path(map, start, end){
57      var queue = [start];
58
59      function insert(x, y){
60
61          if(x < 0 || x >= 100 || y < 0 || y >= 100)
62              return ;
63          if(map[y * 100 + x])
64              return ;
65
66          map[y * 100 + x] = 2;
67          queue.push([x, y]);
68      }
69
70      while(queue.length){
71          let [x, y] = queue.shift();
72          console.log(x, y);
73          if(x === end[0] && y === end[1]) {
74              return true;
75          }
76          insert(x - 1, y);
77          insert(x, y - 1);
78          insert(x + 1, y);
79          insert(x, y + 1);
80
81      }
```



```
function sleep(t){
    return new Promise(function(resolve){
        setTimeout(resolve, t);
    });
}


async function findPath(map, start, end){
    let table = Object.create(map);
    let queue = [start];

    async function insert(x, y, pre){
        if(x < 0 || x >= 100 || y < 0 || y >= 100)
            return ;
        if(table[y * 100 + x])
            return ;

        await sleep(1);
        container.children[y * 100 + x].style.backgroundColor = "lightgreen";
        table[y * 100 + x] = pre;
        queue.push([x, y]);
    }

    while(queue.length){
        let [x, y] = queue.shift();
        console.log(x, y);
        if(x === end[0] && y === end[1]) {
            let path = [];

            while(x != start[0] || y != start[1]) {
                path.push(map[y * 100 + x]);
```