JS第二周第二天

`javascript`

1.声明变量

```
<script>
    var a;//声明
    a=1;//定义

    //不声明，直接定义
    b=10;//相当于直接给window增加属性b，值为10
    console.log("b" in window);//true
    //不可以使用既没有声明，也没有定义的变量
    //console.log(c);//c is not defined

    //let a; 不可以声明已经声明的变量,不管之前是用什么方式声明的变量

    //绑定事件都是绑的函数定义change，不是执行change()
    function change() {
        box.style.backgroundColor="yellow";
    }
</script>
```

2.时间

```
<script>
  let ary = ["星期天", "星期一", "星期二", "星期三", "星期四", "星期五",
"星期六"];

  function fn() {
    //new Date 使用构造函数方式创建
    //获取本地的当前时间
    //let time=new Date  Date后面加不加()都可以
    let time = new Date();//标准时间格式
    //Wed Apr 25 2018 10:20:04 GMT+0800 (中国标准时间)
    //获取四位的年 2018
    let year = time.getFullYear();
    //获取月 0-11
    let month = time.getMonth();
    //获取星期 0-6 星期天-星期六
    let week = time.getDay();
    //获取日
    let day = time.getDate();
    //获取小时
    let hours = time.getHours();
    //获取分钟
    let minutes = time.getMinutes();
    //获取秒
    let seconds = time.getSeconds();
    //2018年4月25日 星期三 10:30:45
    let str = `${year}-${month+1}-${day} ${ary[week]} ${hours}:${minutes}:${seconds}`;
    getDate.innerHTML = str;
  }
  fn();
  setInterval(fn,1000);


  //在${}中可以写简单的运算表达式 比如计算,三元运算符

  let sex=1;
  let str1=`性别:${sex?'男':'女'}`;
  console.log(str1);


</script>
```

3.倒计时

```
<script>
  //目标时间 确定的时间
  //let target=new Date("2018 4 25 12:00:00");
  //let target=new Date("2018/4/25 12:00:00");
  //let target=new Date("2018-4-25 12:00:00");
  //let target=new Date("2018*4*25 12:00:00");
  let target = new Date("2018,4,25 13:00:00");
  console.log(target);
  //Wed Apr 25 2018 12:00:00 GMT+0800 (中国标准时间)
  function fn() {
    let now = new Date;
    //获取时间差 毫秒
    let diff = target - now;
    //console.log(diff);//4034371
    //毫秒变成  小时 分钟 秒
    //从大到小计算 先计算小时 不够一小时变成分钟 不够一分钟变成秒
    let h = Math.floor(diff / (1000 * 60 * 60));
    let m = Math.floor((diff % (1000 * 60 * 60)) / (1000 * 60));
    let s = Math.floor(((diff % (1000 * 60 * 60)) % (1000 * 60)) / 1000);
    t.innerHTML = `距离下课还有: ${h}小时 ${m}分钟 ${s}秒`;
  }
  fn();
  setInterval(fn,1000);
</script>
```

4.数组方法

```
<script>
  //数组原型上的返方法
  console.log(Array.prototype);
  //作用,返回值,参数,原数组是否改变
  //增加
  //1.unshift
  //作用:向数组的前面增加新的项
  //返回值:改变后新数组的length
  //参数:一个或者多个
  //原数组:改变
  var ary = [1, 2, 3];
  var result = ary.unshift("珠峰", "中国");
  console.log(ary);//["珠峰", "中国", 1, 2, 3]
  console.log(result);

  //2.push
  //作用:向数组的后面增加新的项
  //返回值:改变后新数组的length
  //参数:一个或者多个
  //原数组:改变
  var ary = [1, 2, 3];
  var result = ary.push(10, 20);
  console.log(result);//5
  console.log(ary);//
```

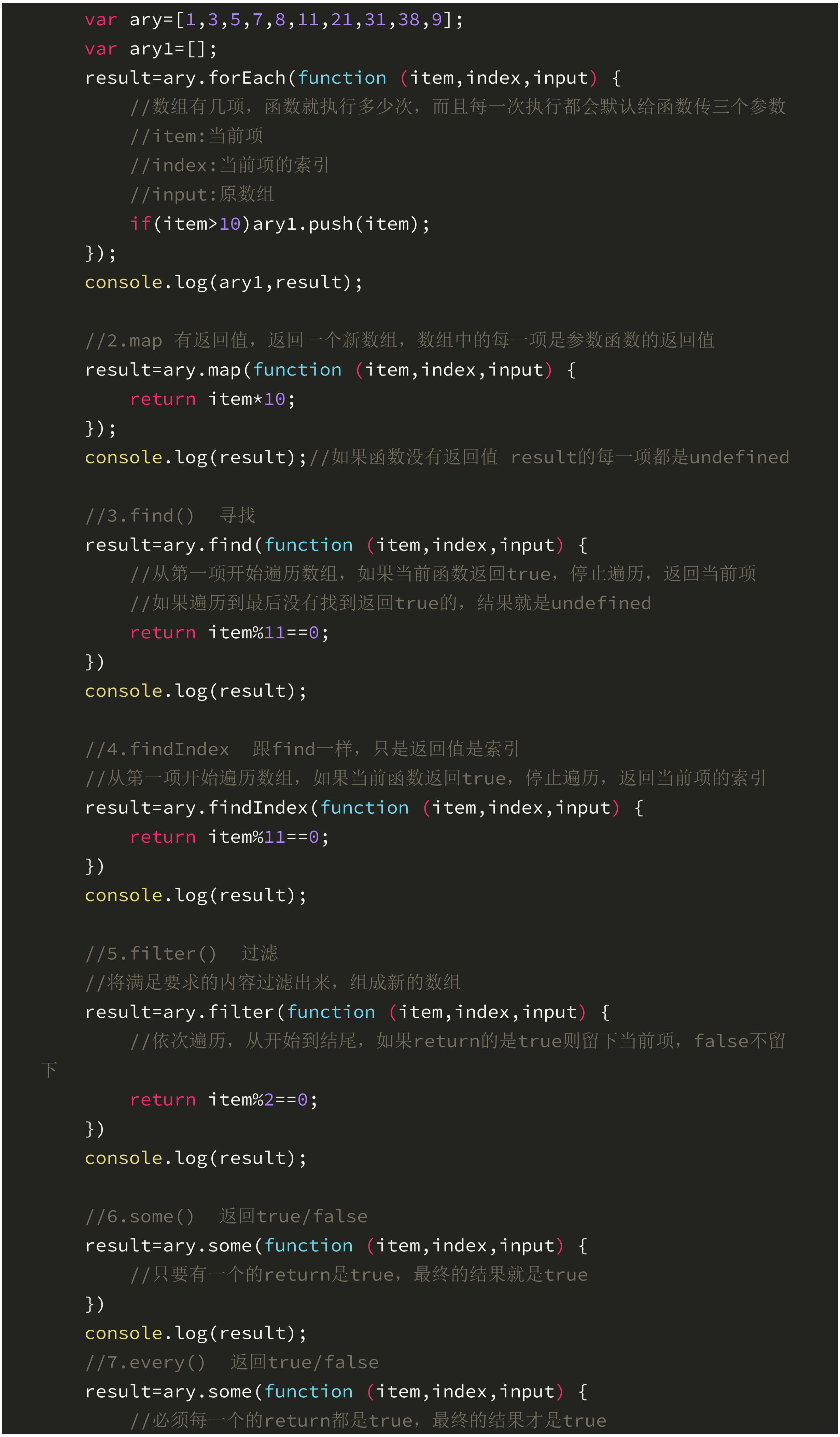
```
//删除
//3.shift()
//作用:删除数组的第一项
//返回值:删除的那一项
//参数:没有
//原数组:改变
var ary = [1, 2, 3];
var result = ary.shift();
console.log(result);//1
console.log(ary);// [2,3]

//4.pop()
//作用:删除数组的最后一项
//返回值:删除的那一项
//参数:没有
//原数组:改变

var ary = [1, 2, 3];
var result = ary.pop();
console.log(result);//3
console.log(ary);// [1,2]

//查找
//5.slice
//作用:从原数组中查找某些项
//返回值:查找出的内容组成的新数组
//原数组:不变
//参数:
// (n,m): 从索引n找到索引m  不包括m  m>n 如果n>m ->[]
// (n) : 从索引n找到最后
// () : 从索引0找到最后  得到一个跟原数组一模一样的新数组
//负数: 倒着数
var ary = [1, 2, 3, 4, 5];
var result = ary.slice(3, 1);
console.log(result);//[]
console.log(ary.slice(-4, 2));//[2]

//6.splice
//作用:截取或者修改数组
//返回值:截取出的内容组成的新数组
//原数组:改变
//参数:
//():->[]
//(n):从索引n截取到最后
//(n,m): 从索引n截取m个
//(n,m,x1,x2,...) : 从索引n截取m个 替换成x1,x2,...
var ary = [1, 2, 3, 4, 5];
//console.log(ary.splice(3, 2));//[4,5]
```

```
    //console.log(ary.splice(3, 2,"js","css","HTML"));//[4,5]
    console.log(ary.splice(3, 0, "哈哈"));
    //(n,0,x1,x2....) 在索引n的前面插入x1,x2....
    console.log(ary);


    //7.concat
    //作用:拼接/合并数组
    //返回值:合并后的新数组
    //原数组:不变
    //参数:既可以是数组也可以是值,个数不规定
    var ary = [1, 2, 3, 4, 5];
    console.log(ary.concat(1, 3, 100));
    console.log(ary.concat([100, 200, 300]));
    console.log(ary.concat([100, 200, 300], 1));
    console.log(ary);

    //8.join
    //作用:将数组按照指定字符拼接成字符串
    //返回值:字符串
    //原数组:不变
    //参数:指定字符 不传跟toString一样
    var ary = [1, 2, 3, 4, 5];
    console.log(eval(ary.join("+")));
    console.log(ary.join());

    //9.reverse()
    //作用:倒序
    //返回值:改变后的新数组
    //原数组:改变
    //参数:不需要
    var ary = [1, 2, 3, 4, 5];
    console.log(ary.reverse());
    console.log(ary);

    //10.indexOf
    //作用:查找某一项在数组中第一次的索引
    //返回值:索引,没有 -1
    //原数组:不变
    //参数
    //(val):某一项
    //(val,index):第二个参数表示开始查找的索引
    var ary = [1, 2, 3, 4, 5, 6, 7, 8];
    console.log(ary.indexOf(4));//3
    console.log(ary.indexOf(4, 4));//-1
    //lastIndexOf
    //作用:查找某一项在数组中最后一次的索引
    //返回值:索引,没有 -1
    //原数组:不变
```

```
  //参数
  //(val):某一项
  //(val,index):第二个参数表示开始查找的索引


  //11.includes()
  //作用:判断数组中有没有某一项
  //返回值:true/false
  //原数组:不变
  //参数
  //(val):某一项 默认从索引0开始找
  //(val,index):第二个参数表示开始查找的索引
  var ary = [1, 2, 3, 4, 5, 6, 7, 8];
  console.log(ary.includes(3));
  console.log(ary.includes(3, 4));//false

  //12.fill()
  //作用:填充数组
  //返回值:改变后的新数组
  //原数组:改变
  //参数:
  //(val):将数组的每一项都变成val
  //(val,n):从索引n开始到最后全都变成val
  //(val,n,m):从索引n到索引m(不包括m)都变成val

  var ary = [1, 2, 3, 4, 5, 6, 7, 8];
  console.log(ary.fill("哈哈", 3));
  console.log(ary);

  //13.copyWithin
  //作用:从数组本身copy某一部分 去覆盖某一部分
  //返回值:改变后的新数组
  //原数组:改变
  //参数:见图
  var ary = [1, 2, 3, 4, 5, 6, 7, 8];
  console.log(ary.copyWithin(3, 5));//[1, 2, 3, 6, 7, 8, 7, 8]
  var ary = [1, 2, 3, 4, 5, 6, 7, 8];
  console.log(ary.copyWithin(3, 5, 6));//[1, 2, 3, 6, 5, 6, 7, 8]
  var ary = [1, 2, 3, 4, 5, 6, 7, 8];
  console.log(ary.copyWithin(6, 3, 7));
</script>
```

5.遍历数组的方法

```
<script>
    //1.forEach  没有返回值,原数组不改变
```



```
    var ary=[1,3,5,7,8,11,21,31,38,9];
    var ary1=[];
    result=ary.forEach(function (item,index,input) {
        //数组有几项，函数就执行多少次，而且每一次执行都会默认给函数传三个参数
        //item:当前项
        //index:当前项的索引
        //input:原数组
        if(item>10)ary1.push(item);
    });
    console.log(ary1,result);

    //2.map 有返回值，返回一个新数组，数组中的每一项是参数函数的返回值
    result=ary.map(function (item,index,input) {
        return item*10;
    });
    console.log(result);//如果函数没有返回值 result的每一项都是undefined

    //3.find()  寻找
    result=ary.find(function (item,index,input) {
        //从第一项开始遍历数组，如果当前函数返回true，停止遍历，返回当前项
        //如果遍历到最后没有找到返回true的，结果就是undefined
        return item%11==0;
    })
    console.log(result);

    //4.findIndex  跟find一样，只是返回值是索引
    //从第一项开始遍历数组，如果当前函数返回true，停止遍历，返回当前项的索引
    result=ary.findIndex(function (item,index,input) {
        return item%11==0;
    })
    console.log(result);

    //5.filter()  过滤
    //将满足要求的内容过滤出来，组成新的数组
    result=ary.filter(function (item,index,input) {
        //依次遍历，从开始到结尾，如果return的是true则留下当前项，false不留
下
        return item%2==0;
    })
    console.log(result);

    //6.some()  返回true/false
    result=ary.some(function (item,index,input) {
        //只要有一个的return是true，最终的结果就是true
    })
    console.log(result);
    //7.every()  返回true/false
    result=ary.some(function (item,index,input) {
        //必须每一个的return都是true，最终的结果才是true
    })
    console.log(result);

    //8.reduce 迭代
    //[1,3,5,7,8,11,21,31,38]求和
    //第二个参数表示初始值
    result=ary.reduce(function (prev,item) {
        //prev:上一次执行函数的返回值
        //item:当前项
        return prev+item;
    },1000);
    console.log(result);
    //9.reduceRight 从右往左迭代

    //10.sort()  排序   原数组改变
    //ary.sort() 只能排序10以下的
    ary.sort(function (a,b) {
        //return a-b;//升序
        return b-a;//降序
    });
    console.log(ary);

    //11.splice（i,1） 删除数组索引为i的那一项
</script>
```

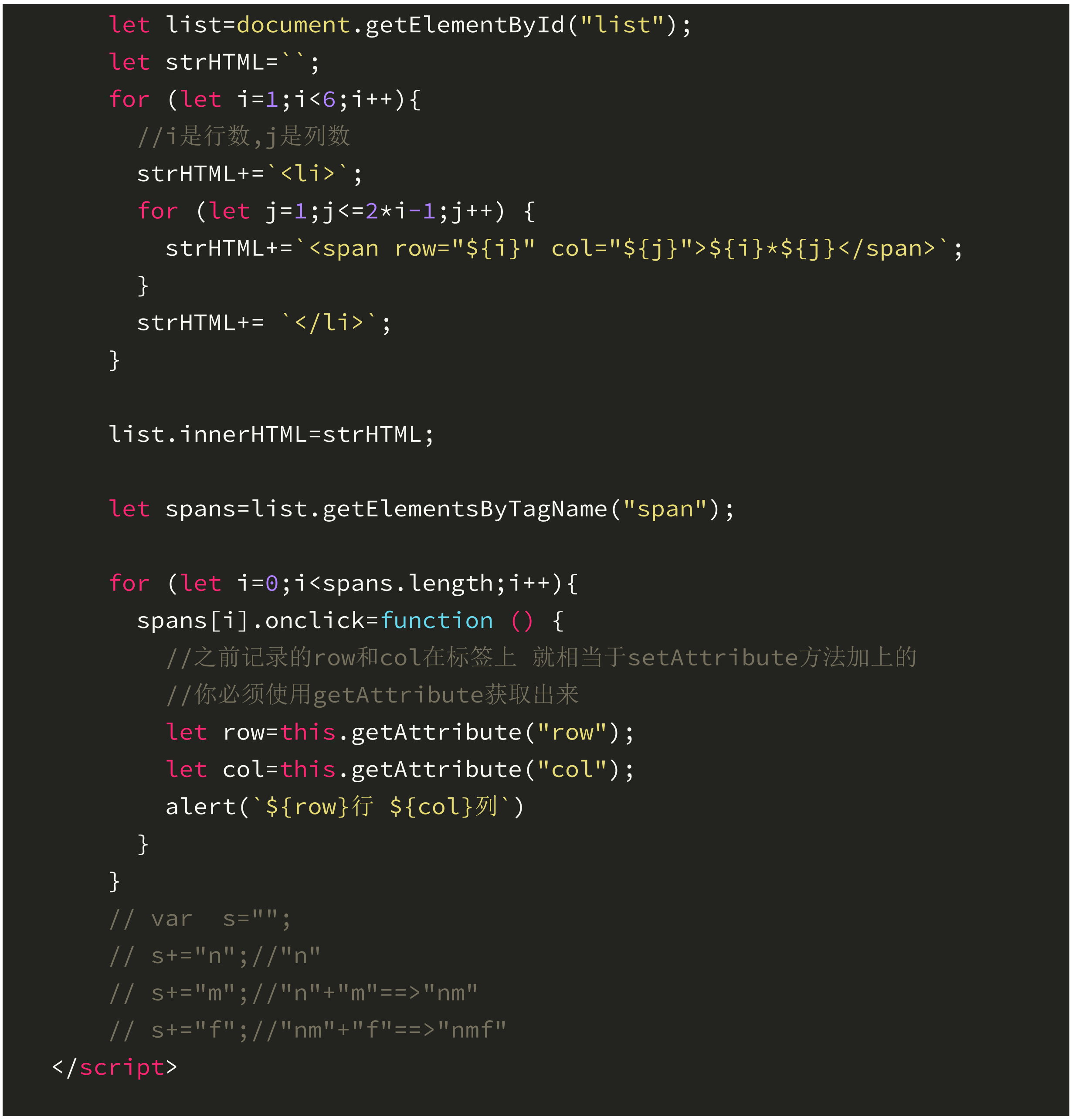
6.数组练习题

```
<script>
    var arr1=[{name:"AA",age:12},{name:"FF",age:31},{name:"NN",age:19},{name:"PP",age:26},{name:"SS",age:52}];

    //1.找出数组中的name是"PP"的将其age变成24

    // for (var i=0;i<arr1.length;i++){
    //   if(arr1[i].name=="PP"){
    //     arr1[i].age=24
    //   }
    // }

    // for (let [index,item] of arr1.entries()){
    //   if(item.name=="PP"){
    //     arr1[index].age=24
    //   }
    // }

    // arr1.forEach(function (item,index) {
```

```
    //   if(item.name=="PP"){
    //     arr1[index].age=24;
    //   }
    // });

    var index=arr1.findIndex(function (item,index) {
      return item.name=="PP";
    });
    //arr1[index].age=24;

    //2.找出数组中的name是"PP"的将其删除
    //splice(i,1):删除数组索引为i的那一项
    //arr1.splice(index,1);
    //console.log(arr1);

    //3.找出数组中的年龄大于30的组成新的数组

    var arr2=arr1.filter(function (item,index) {
      return item.age>30;
    });
    console.log(arr2);

</script>
```

7.面试题

```
<style>
        #list li{
            text-align: center;
            list-style: none;
        }
        #list li span{
            display: inline-block;
            width: 100px;
            height: 100px;
            background-color: #6472ff;
            margin: 10px;
        }
</style>


<body>
    <ul id="list"></ul>
</body>

<script>
```



```
    let list=document.getElementById("list");
    let strHTML=``;
    for (let i=1;i<6;i++){
      //i是行数,j是列数
      strHTML+=`<li>`;
      for (let j=1;j<=2*i-1;j++) {
        strHTML+=`<span row="${i}" col="${j}">${i}*${j}</span>`;
      }
      strHTML+= `</li>`;
    }

    list.innerHTML=strHTML;

    let spans=list.getElementsByTagName("span");

    for (let i=0;i<spans.length;i++){
      spans[i].onclick=function () {
        //之前记录的row和col在标签上 就相当于setAttribute方法加上的
        //你必须使用getAttribute获取出来
        let row=this.getAttribute("row");
        let col=this.getAttribute("col");
        alert(`${row}行 ${col}列`)
      }
    }
    // var  s="";
    // s+="n";//"n"
    // s+="m";//"n"+"m"==>"nm"
    // s+="f";//"nm"+"f"==>"nmf"
</script>
```